LOW SENSITIVITY DIGITAL FILTERS

A Thesis Submitted
In Partial Fulfilment of the Requirements
for the Degree of

MASTER OF TECHNOLOGY

by

SANJEEV KUMAR RASTOGI

to the

DEPARTMENT OF ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

APRIL, 1987

CERTIFICATE

Certified that the work embodied in this thesis entitled 'Low Sensitivity Digital Filters' has been carried out by Mr. Sanjeev Kumar Rastogi under my supervision and the same has not been submitted elsewhere for a degree.

(V. P. Sinha)
Professor
Department of Electrical Engineer
Indian Institute of Technology
KANPUR.

ACKNOWLEDGEMENTS

I am greatly indebted to my supervisor, Dr. V.P. Sinha for introducing me to the field of Digital Filters with Low Sensitivity. Accomplishment of the present work would not have been possible without his sustained encouragement and open hearted cooperation.

Finally I express my sincere appreciation for the neat typing by Mr. L.S. Bajpai.

Sanjeev Kumar Rastogi

CONTENTS

LIST OF FIGURES

Figure No.		Page

ABSTRACT

Low sensitivity digital filters are synthesized both by direct and indirect methods. Several classes of these filters are well known. It is also observed that their structures bear close similarity with one another. Wave digital filters, lattice digital filters, LBR 2-pair based filters and orthogonal filters are all considered and their design procedures are explained with illustrative examples. A detailed review of low sensitivity digital filters is presented from a unified point of view in which the notion of orthogonality plays a central role. The work is aimed at giving to those not familiar with the the subject, an insight into the theory and design of these filters. No prior background in analog filter theory is assumed. To maintain flow in the text, details of mathematical analysis are given only if essential for gaining insight. Recent work on VLSI implementation of these filters is also discussed.

INTRODUCTION

Sensitivity is an important consideration for the design and implementation of digital filters. In hardware applications, low sensitivity to coefficient values is the designer's primary concern. This is so because cost of the filter can be reduced by using registers of minimal length to store its parameters. Similarly for the VLSI implementation, filter structures that are modular in nature as well as numerically stable are generally preferred. It is known that the conventional schemes, based on factorization and partial fraction expansion of the transfer function, do not yield structurds with these desirable properties.

Several methods have been suggested in the past for the design of low sensitivity digital filters. An indirect realisation scheme based on the design of doubly terminated LC filters was proposed by Fettweis [1]. This method gives the so called wave digital filters. Their low sensitivity is explained using the concept of pseudolossless [2] elements analogous to the concept of lossless analog elements. Later, Gray and Markel gave an algorithm that generates a cascade of lattice structures from the given transfer function directly in the digital domain [3]. In this algorithm, the transfer function is realised by tapping outputs of the lattice building blocks whose transfer matrices are orthogonal [4]. The concept

of orthogonality was later used by Dewilde et. al. to design cascaded orthogonal digital filters [5]. This method of realisation is based on a direct factorization of a lossless transfer matrix. Another scheme was proposed by Vaidyanathan and Mitra [6] which generates a cascade of lossless bounded two-pairs extracted from the given transfer function. Work is also being done to investigate the design of low sensitivity structures that are suitable for VLSI implementation of digital filters [7].

A vast amount of literature is available on these realisation schemes. Since all of these methods give low sensitivity implementations of digital filters, a need is felt to evolve a common theory about them. The ideas, scattered in the literature in different forms, tend to create the impression that each of these methods gives an entirely new low sensitivity structure. A closer look however reveals that they can be brought together under a unified theoretical framework [8].

The aim of this thesis is to put the various ideas together to give an integrated approach for designing low sensitivity digital filters. The entire presentation is of a tutorial nature with proper examples included for illustration. An effort is made here to give the reader a compact yet complete picture of the recent trends in the theory and design of these filters.

In Section 1, we discuss the theory of doubly terminated analog LC filters and microwave filters, which is needed to build the base for an understanding of wave digital filters, described in Section 3. Section 2 explains the idea of low sensitivity for analog filters, and also includes a general discussion on the theory of low sensitivity digital filters. In Section 4, we explain one by one various direct realisation schemes for low sensitivity digital filters namely : lattice digital filters, orthogonal digital filters and LBR two-pair structures. Later in Section 5, we discuss a unified theory that explains the similarity between all low sensitivity structures. We also discuss how the orthogonality property of a prototype building block is useful for its stable implementation by means of CORDIC technique. For sake of completeness a brief discussion of the issues related to VLSI design of low sensitivity digital filters is also included. This is done in view of interesting new structures that are nowadays being investigated to take advantage of the possibilities of parallel data processing.

SECTION ONE

REVIEW OF ANALOG FILTERS

1.1 INTRODUCTION

In the modern era of digital processing of signals, digital filters have replaced the conventional analog filters in most of the applications. Several classes of digital filters have very little in common with the analog filters. As a result, the well established theory of analog filters has been relegated to the background. It is felt, however, that the theory of low-sensitivity digital filters is difficult to understand without a prior knowledge of the class of analog filters with similar properties.

Keeping this difficulty in mind we first introduce in this section some of the basic concepts related to the analog filter theory [9]. We consider the design aspects of doubly terminated lossless filters that are known to have low sensitivity with respect to the element values [10]. We illustrate the ladder and lattice network realisations of these filters [11]. Finally, we introduce some fundamental aspects of microwave filters. These filters have a complete theory of their own. We consider only the class of microwave filters that can be synthesized analogously to the doubly terminated LC filters [12]. The reason for including microwave filters here is that they constitute a convenient pedagogic link between analog and digital filters.

1.2 DEFINITIONS

The most common analog filters with low sensitivity properties are the insertion-loss filters consisting of a lossless 2-port network terminated at the output by a load resistance R_2 (Fig. 1.1). The 2-port network is driven by a voltage source V_S having a source resistance R_S.

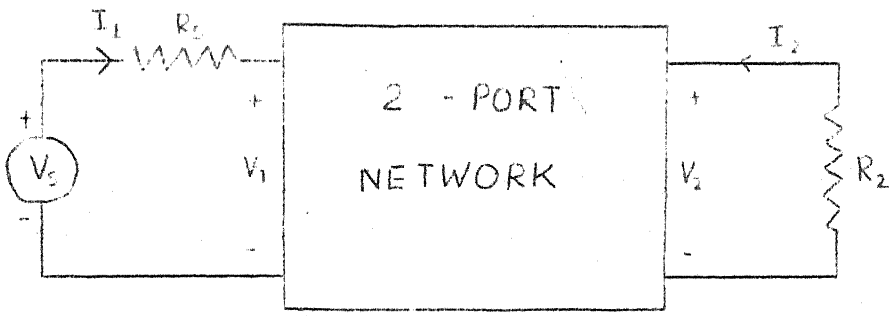


Fig. 1.1 A two-port network inserted between resistive terminations.

The following are a number of standard terms for analog filters.

Transfer Function

The transfer function H(s) of a filter is defined as the ratio of an output port variable to an input port variable. These variables can be either voltages or currents. Thus, if only voltage variables are considered, then the transfer function of the 2-port in Fig. (1.1) is given by

$$H(s) = \frac{V_2(s)}{V_S(s)}$$

Transmittance and Reflectance

A lossless 2-port network does not absorb any real power. Power is absorbed only by the source and the load resistances. According to the maximum power transfer theorem, the power transmitted to the load has an upper bound, say P_{2MAX}. The transmitted power $P_2(\omega)$ at any frequency, ω, can be normalised w.r.t. this upper bound. This defines a new function t(s), known as the transmittance. It is given by

$$t(j\omega) = \frac{P_2(\omega)}{P_{2MAX}} = 2\sqrt{\frac{R_S}{R_2}}\ H(j\omega)\,,\quad 0 \leq |t(j\omega)| \leq 1$$

where previous notations are used.

Reflectance of a 2-port network is a function which is complementary to the transmittance. It is denoted by $\rho(s)$. The relationship between transmittance and reflectance is expressed as

$$t(s)\ t(-s) + \rho(s)\rho(-s)\Big|_{s=j\omega} = 1 \tag{1.1}$$

Loss function of the filter

The filter characteristic is conveniently expressed by the loss function $L(\omega^2)$ and another function $K(s)$. These functions are related to $t(s)$ by

$$L(\omega^2) = 20\ \mathrm{Log}\left(\frac{1}{|t(j\omega)|}\right) = 10\ \mathrm{Log}(1 + |K(j\omega)|^2)$$

The loss function is expressed in decibels. Further, if $t(s)$ and $K(s)$ are of the form

$$t(s) = \frac{p(s)}{e(s)} \quad \text{and} \quad K(s) = \frac{f(s)}{p(s)}$$

where $p(s)$, $e(s)$ and $f(s)$ are polynomials in s and are related by the equality

$$e(s)\ e(-s) = p(s)\ p(-s) + f(s)\ f(-s)\Big|_{s=j\omega} \tag{1.2}$$

then

$$\rho(s) = f(s)/e(s).$$

Eqn. (1.2) is often called the Feldkeller equation. Several generalised filter functions viz. the Butterworth or the Chebyshev function can be chosen for $K(s)$ so that all the design specifications are properly met (A.1).

Driving point functions

The ratio of the input voltage to the input current is defined as the driving point impedance function $Z_{IN}(s)$. Its

reciprocal is another function called the <u>driving point admittance function</u> $Y_{IN}(s)$. An analog network is realised using these immittance (impedance or admittance) functions, as it will be seen later. A necessary and sufficient condition for the realisability of Z_{IN} or Y_{IN} is that it be positive real (A.2).

The driving point impedance $Z_{IN}(s)$ is also related to the reflectance. It is given by

$$Z_{IN}(s) = R_S\left(\frac{1 \pm \rho}{1 \mp \rho}\right) = R_S\left(\frac{f \pm e}{f \mp e}\right) \tag{1.3}$$

1.3 LADDER NETWORK FILTER SYNTHESIS

The immittance function obtained from eqn. (1.3) can be realised by several alternative methods. The most common ones are the so called Foster's and Cauer's realisations (A.3). The latter is also known as the ladder synthesis. To illustrate this method, we now consider a simple example of filter design.

EXAMPLE 1.1

Design a normalised Chebyshev filter with the following specifications

passband : attenuation $\leq$ 0.96 dB, Cut off frequency = 1 rad/sec.

stopband : attenuation $\geq$ 30 dB, Cut off frequency = 4 rad/sec.

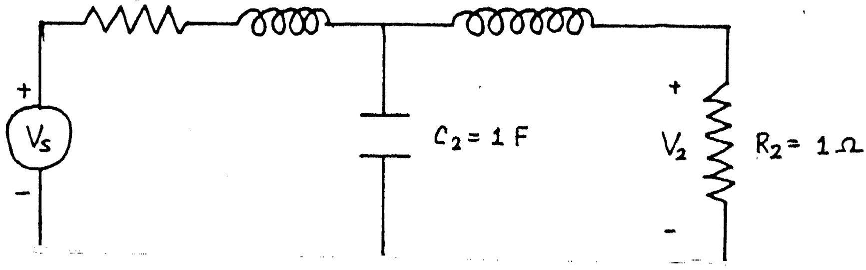


Fig. 1.2 Third order Chebyshev lowpass filter of Example (1.1).

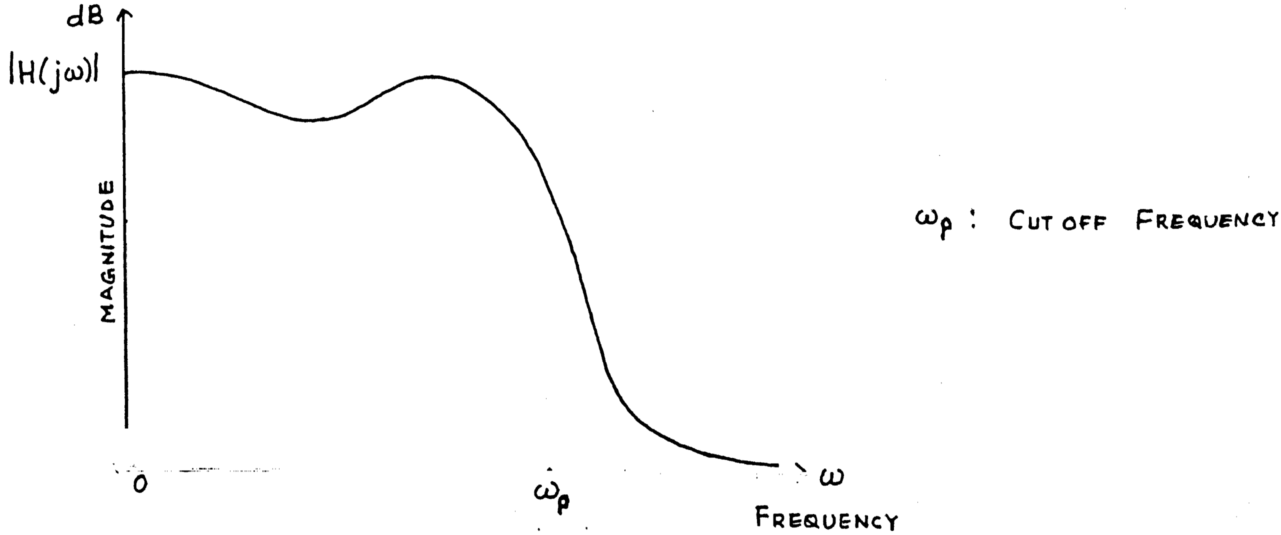


Fig. 1.3 Frequency response of the lowpass filter of Example (1.1).

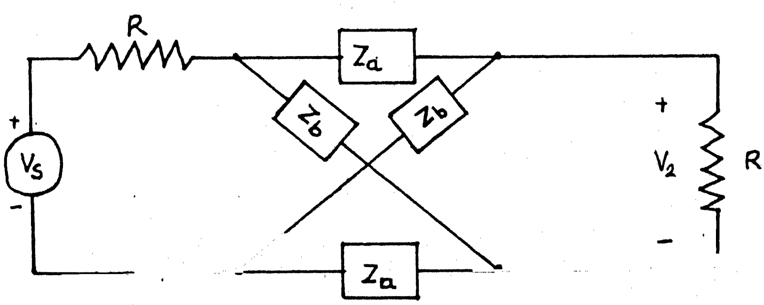


Fig. 1.4 A symmetric lattice filter network.

If passband requirements are exactly met for the Chebyshev function, using eqns. (A.3) and (A.4) we obtain $n \geq 2.97$ (= 3) and $C = 0.5$.

Solving the Feldtkeller eqn. (1.2) we obtain

$$f = 2s^3 + 1.5\,s$$

$$p = 1$$

$$e = 2s^3 + 2s^2 + 2.5\,s + 1$$

By using eqn. (1.3) we get $Z_{IN}(s)$ as

$$Z_{IN}(s) = \frac{e+f}{e-f} = \frac{4s^3 + 2s^2 + 4s + 1}{2s^2 + s + 1}$$

When realised by the ladder method as in (A.3), we get

$$L_1 = 2H,\ C_2 = 1F,\ L_3 = 2H,\ R_1 = R_2 = 1\ \Omega$$

the corresponding network is shown in Fig. (1.2) and its frequency response is plotted in Fig. (1.3).

1.4 SYMMETRIC LATTICE FILTERS

Lattice realisation of the filters is also very common, especially when resonant circuits are involved. A symmetric lattice filter is represented by two lossless impedance functions $Z_a(s)$ and $Z_b(s)$ as shown in Fig. (1.4)

If the source and the load resistances are equal then it can be shown (A.4) that

$$K(s) = \frac{Z_a Z_b - R^2}{R(Z_b - Z_a)} \tag{1.4}$$

and,

$$U(s) = \frac{1}{t(s)} = \frac{(Z_a + R)(Z_b + R)}{R(Z_b - Z_a)} \tag{1.5}$$

where U and K are related by

$$U(s)\ U(-s) = 1 + K(s)\ K(-s)$$

Solving eqns. (1.4) and (1.5) for Z_a and Z_b which are lossless and hence odd functions in s,

$$Z_a = R\left(\frac{U_e - 1}{U_o + K_o}\right) \tag{1.6a}$$

$$Z_b = R\left(\frac{U_e + 1}{U_o - K_o}\right) \tag{1.6b}$$

Subscripts e, o denote even,odd parts respectively. The impedance functions are realised by any of previously mentioned methods (A.3). We now consider an example of lattice filter synthesis.

<u>EXAMPLE 1.2</u>

Design a 3rd order Chebyshev lattice filter with specifications as in example 1.1 .

Using eqn. (A.8) we obtain, similar to e.g. 1.1

$$K(s) = 2s^3 + 1.5\ s$$

$$U(s) = 2s^3 + 2s^2 + 2.5\ s + 1$$

Separating even and odd parts we get,

$$K_o = 2s^3 + 1.5\ s, \quad U_o = 2s^3 + 2.5\ s, \quad U_e = 2s^2 + 1$$

Eqns. (1.6a) and (1.6b) give

$$Z_a = 2s \quad \text{and} \quad Z_b = 2s + 2/s$$

The complete lattice network is shown in Fig.(1.5).

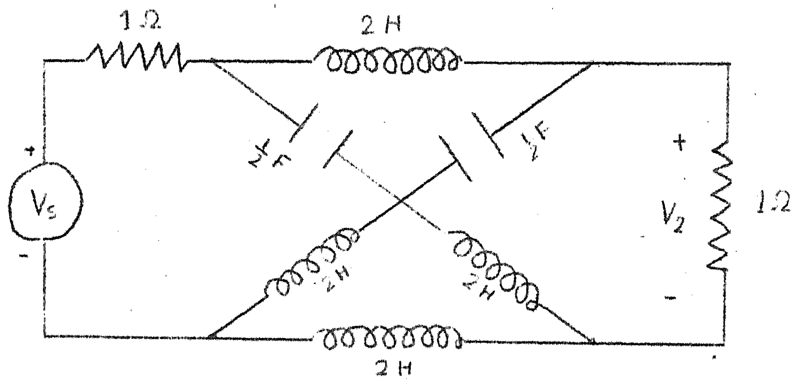


Fig. 1.5 Lowpass lattice filter of Example (1.2).

1.5 MICROWAVE FILTERS

For microwave applications, we require filters that have low sensitivity and are also suitable for high frequency operations. In such applications, instead of using lumped elements such as L and C. We use short circuited and open circuited transmission lines. In contrast to the lumped ones, these elements are distributed in nature. Although the immittance of these distributed elements involve exponential functions [12], they can be converted into rational functions by using the so called Richards' transformation. The design, therefore, can be carried out in an equivalent domain.

Richards' Transformation

This is a transformation from the s-domain to the ψ-domain, where

$$\psi = \tanh(sT/2), \text{ where}$$

T = 2l/v = Round trip delay in the transmission line of length l

v = velocity of propogation

The inverse transformation is

$$e^{sT} = \frac{1+\psi}{1-\psi}$$

Thus the exponentials are transformed to rational functions. It is also observed (A.5) that all the conditions

of stability are preserved in the ψ-domain also.

Unit Element

In microwave theory an important element is the 2-port element called the <u>unit element</u>, which has the following transfer matrix

$$\begin{bmatrix} v_1 \\ v_2 \end{bmatrix} = \frac{Z_o}{\psi} \begin{bmatrix} 1 & \sqrt{1-\psi^2} \\ \sqrt{1-\psi^2} & 1 \end{bmatrix} \begin{bmatrix} i_1 \\ i_2 \end{bmatrix}$$

It is symbolized as shown in Fig.(1.6). Its importance will be discussed later. First, we will consider some design aspects of microwave filters.

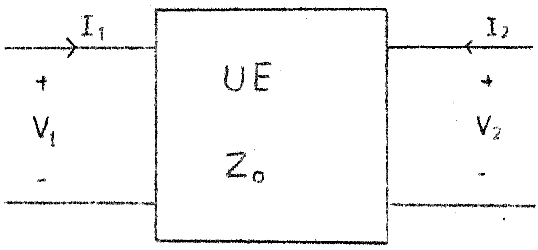
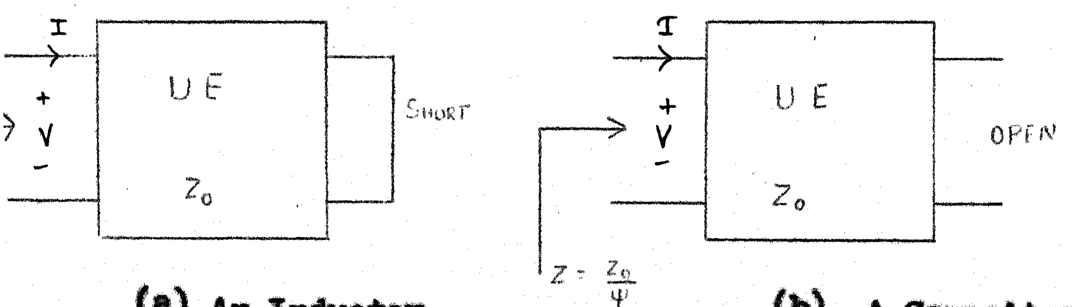


Fig. 1.6 A unit element.



(a) An Inductor (b) A Capacitor

Fig. 1.7 Lumped and distributed element correspondence.

Design Considerations

The design techniques for lumped element filters can be straight-forwardly carried over to the microwave filter design by virtue of Richard's transformation. In this method, L and C elements of the former case are replaced by short circuited and open circuited lines respectively in a manner that is shown in Fig. 1.7(a) and 1.7(b). This method, however, has one drawback. In the microwave filter design, all elements can be connected only in parallel.

To overcome this limitation, we can use Kuroda's identities to convert a series connection into a parallel one [12]. One such identity is shown in Fig. (1.8).

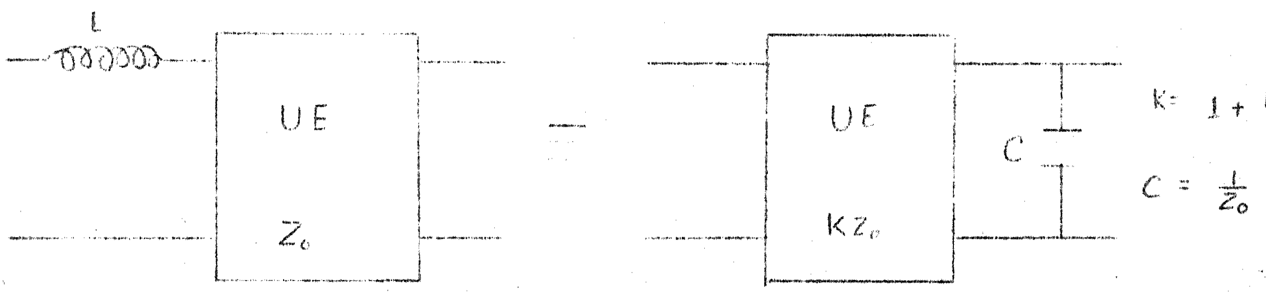


Fig. 1.8 An example of Kuroda's identity.

Unit Element Extraction

One can avoid the use of Kuroda's identity, which is a cumbersome process [13]. Whenever a shunt element cannot be extracted, we extract a unit element from the immittance function, which is a PRF. A unit element extraction is done using the Richards' theorem (A.6), which leaves the remainder again a PRF but of a lower degree as shown in Fig. (1.9).

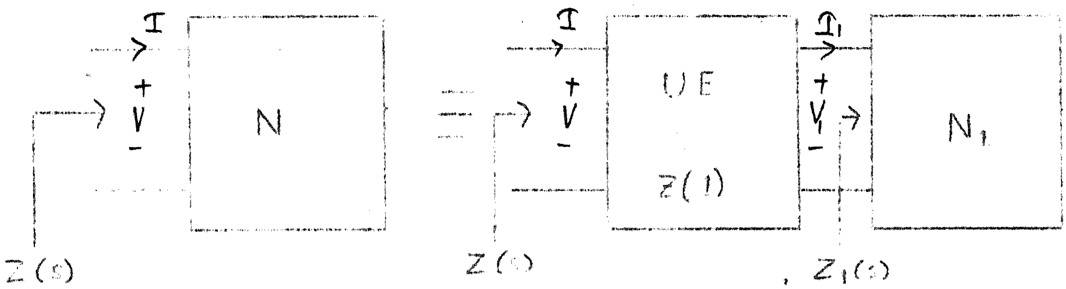


Fig. 1.9 Schematic illustration of Richard's theorem.

The complete procedure is best illustrated by an example.

EXAMPLE 1.3

Design a microwave filter with same specifications as in example 1.1. Our aim is to synthesize the immittance function in as few steps as possible.

We would like to extract a capacitance first. This can be done if we choose the lower signs in eqn. (1.3) to get

$$Y(s) = \frac{4s^3 + 2s^2 + 4s + 1}{2s^2 + s + 1}$$

$$= 2s + Y_1(s)$$

where

$$Y_1(s) = \frac{2s+1}{2s^2+s+1}$$

Since a capacitance has already been extracted, now we have to extract a unit element. Noting that

$$Y_1(1) = 3/4,$$

we extract a unit element with $Z_o = 4/3$. This leaves $Y_2(s)$ as

$$Y_2(s) = \frac{18s^2 + 27s + 12}{8s + 12}$$

$$= 9/4s + Y_3(s)$$

where

$$Y_3(s) = \frac{3}{2s+3} \quad \text{and} \quad Y_3(1) = 3/5$$

Once again, by extracting a unit element with $Z_o = 5/3$, we get

$$Y_4(s) = 2/5s + 1$$

The complete filter network is shown in Fig. (1.10).

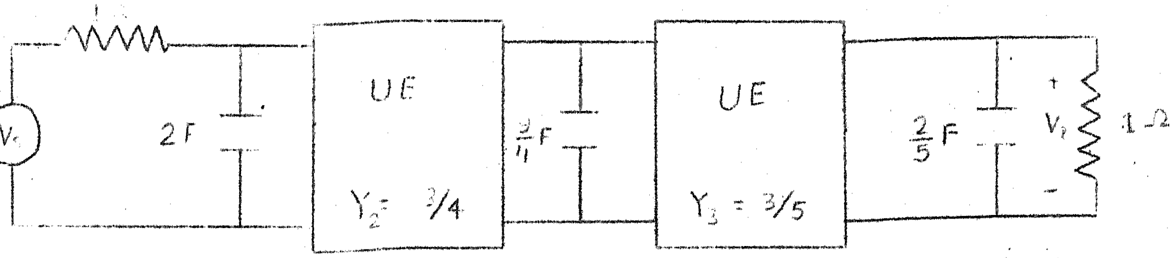


Fig. 1.10 Microwave filter of Example (1.3).

SECTION TWO

LOW SENSITIVITY OF FILTERS : IMPORTANT CONSIDERATIONS

2.1 INTRODUCTION

In the previous section, we familiarised ourselves with some of the important classes of analog filters. Such filters are able to approximate the ideal loss characteristic in an efficient manner. Besides this, we would also like to see that the practical circuit is such that small changes in the element parameters lead to only small changes in the performance characteristic. Two different circuits realising the ideal characteristic may have different sensitivities. It is the designer's task to choose one that has the least sensitivity. Thus sensitivity is an important consideration for the design of any kind of filters.

In the discussions that follow, we first see how the notion of 'sensitivity' is mathematically quantified. This would help us understand the low sensitivity of analog doubly terminated lossless filters in precise terms. We will also introduce the concept of 'structural' and 'lossless' boundedness. These notions are needed to explain a general theory about the low sensitivity of digital filters that has been evolved in recent years. Finally, we touch upon the vector space model of signals. The latter is important today due to the concept of parallel data processing which is now

possible through VLSI implementations of digital filters.

2.2 DEFINITION OF SENSITIVITY

If H denotes the transfer function of a filter and p is one of the design parameters, then the sensitivity of H w.r.t. p is given by [14].

$$S_p^H = \frac{\partial \ln H}{\partial \ln p} = \frac{\partial H/H}{\partial p/p} \tag{2.1}$$

In other words, it is the ratio of fractional change in value of the transfer function to that of the design parameter. Sensitivity is a figure of merit for any filter structure.

In case of analog filters, the performance characteristic should not be affected much by element values due to their tolerances. In case of digital filters, tolerance is not a problem because multipliers have a predetermined values which is stored in the memory registers. However, these registers can have only a finite word length. This results in the quantization of these multipliers. Thus we have at our disposal only a finite number of realisable transfer functions because the set of parameter values is also finite. These transfer functions should have low sensitivity, if the multipliers are to be implemented with only finite precision.

2.3 LOW SENSITIVITY OF LOSSLESS FILTERS

It was seen earlier [15] that in the case of doubly

terminated lossless filters, the source delivers its maximum available power (MAP) at certain frequencies, if K(s) is a selective function viz. Chebyshev, Butterworth etc. These MAP frequencies, shown in Fig. (2.1) by ω_k, force the low sensitivity in the passband region. Before we state the reason, we would like to illustrate that the sensitivity is precisely zero at these frequencies.

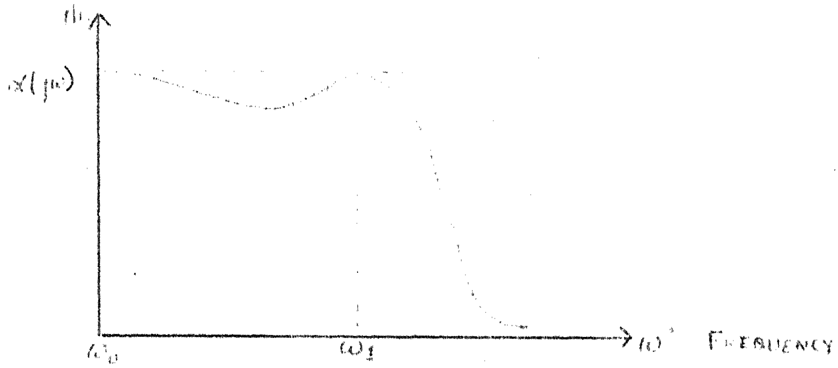


Fig. 2.1 MAP frequencies in the passband.

EXAMPLE 2.1

Network of Fig. (2.2) is a 3^{rd} order Chebyshev filter whose response is shown in Fig. (2.1). Its MAP frequencies are $\omega_o = 0$ and $\omega_1 = \sqrt{(2L-C)}/L\sqrt{C}$ respectively. Show that it has zero sensitivity at these frequencies.

Transmittance t(s) is given by

$$t(s) = \frac{2}{s^3L^2C + 2s^2LC + s(2L+C) + 2}$$

Define $|t(j\omega)|^2 = \alpha(j\omega)$. Using eqn. (2.1) we can show that,

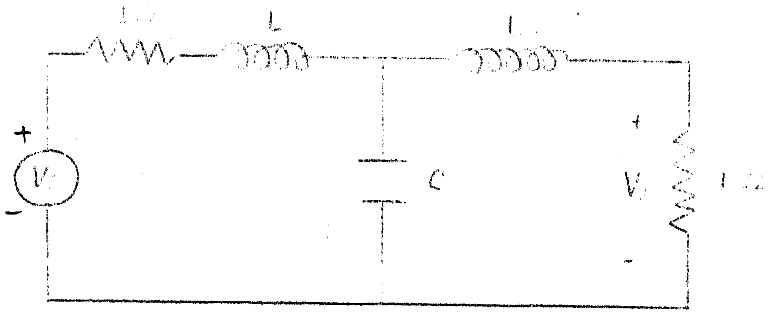


Fig. 2.2 Third order Chebyshev filter of Example (2.1)

$S_x^{\alpha} = 2S_x^{|t(j\omega)|}$, where x is either L or C (element values).

$\alpha(j\omega)$ is given by

$$t(s)\ t(-s)\Big\|_{s=j\omega} = \frac{4}{(1+\omega^2L^2)(\omega^2C^2+\omega^4L^2C^2-4\omega^2LC+4)}$$

taking derivative of α w.r.t. L, we get

$$\frac{\partial\alpha}{\partial L} = \frac{-4\omega^2[\omega^4(L^3C^2-\omega^2LC(3L-C)+(2L-C)]}{(1+\omega^2L^2)^2(\omega^2C^2+\omega^4L^2C^2-4\omega^2LC+4)^2} \qquad (2.2)$$

Substituting for $\omega = \omega_o$ and using eqns. (2.1) and (2.2)

$$S_L^{\alpha}(\omega_o) = 0$$

Similarly substituting for $\omega = \omega_1$ we again get zero sensitivity. Zero sensitivity with respect to C can also be shown.

In connection with the need/for low sensitivity, an important constraint to be imposed on both digital and analog

filters, is that of boundedness. We introduce this concept in the following subsection before discussing the general viewpoint of low sensitivity.

2.4 BOUNDEDNESS OF TRANSFER FUNCTION

As an illustration, we consider a digital infinite impulse response filter having transfer function as

$$H(z) = \frac{a_o + a_1 z^{-1} + \dots a_N z^{-N}}{1 + b_1 z^{-1} + \dots . b_N z^{-N}} , \quad |H(e^{j\omega})| \leq 1$$

By <u>structural boundedness</u> of H(z) we mean that as long as the values of multiplier coefficients, denoted by m_i, are within a permitted range, magnitude of H(z) is bounded above by unity. This boundedness can also be of another type. Consider for example, the following transfer function :

$$H(z) = \frac{a_o + a_1 z^{-1} + \dots . + a_{N-1} z^{-(N-1)} + z^{-N}}{1 + a_{N-1} z^{-1} + \dots + a_1 z^{-(N-1)} + a_o z^{-N}}$$

Here, the numerator polynomial is just the mirror image of the denominator polynomial. As a result, the magnitude of H(z) is always unity. Therefore no matter what changes are made in any of the multiplier, it is always a bounded function. This is known as the <u>lossless boundedness</u>. This kind of boundedness is similar to that of lossless functions in the analog case.

We will now see how the boundedness of the transfer function can induce the low sensitivity property in the filter transfer function.

2.5 LOW SENSITIVITY OF BOUNDED FUNCTIONS

Let us suppose that the magnitude of the transfer function H(z) is precisely unity at certain frequency ω_k for an infinite precision implementation of the multipliers. If now a multiplier m_i is quantized, then the magnitude of transfer function can not increase. This is because the boundedness is 'ensured'. As a result, the plot of H(z) versus m_i is concave downwards as shown in Fig. (2.3).

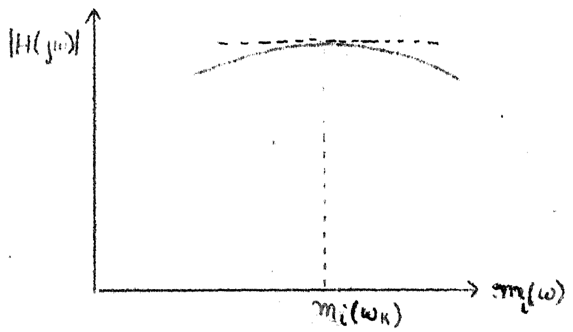


Fig. 2.3 Illustration of low passband sensitivity.

Clearly one can see that

$$\left.\frac{\partial H(e^{j\omega})}{\partial m_i}\right|_{\omega_k} = 0$$

therefore using eqn. (2.1) we get $S^{H}_{m_i} = 0$ at $m_i = m_i(\omega_k)$.

In other words, boundedness of transfer function can force low sensitivity in its implementation. In case of doubly terminated lossless filters, transmittance is a bounded function. This is so because maximum power is transmitted only at MAP frequencies. More is the number of MAP frequencies in the passband, lower will be the sensitivity of the transfer

function. Structural boundedness is referred more commonly as <u>structural passivity</u> in analog case.

So far we have considered situations where there is only a single input signal (or sequence) and a corresponding single output signal (or sequence). In digital filters, it is desirable that multi-input sequences too can be handled. This is in order to maximise the 'active' time of each processor unit especially when we consider the VLSI implementation. In the following subsection we examine this more general situation.

2.6 SIGNAL SPACE AND STRUCTURAL BOUNDEDNESS

From now onwards in this section, we consider signals as a member of n-dimensional vector space. Correspondingly, the transfer function is now a n-port transfer matrix. This n-port structure is shown in Fig. (2.4).

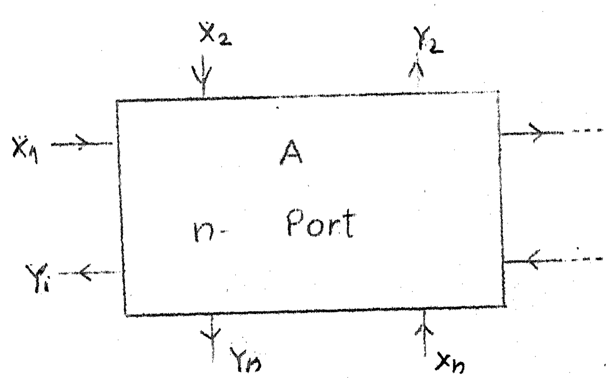


Fig. 2.4 A generalised digital N-port structure.

Each port (i) is represented by an input sequence $x_i(n)$ and output sequence $y_i(n)$. The input-output relationship can

also be looked upon as a linear transformation T from the input-vector space X to the output vector space Y. As an illustration for a 2-port case, T is a 2x2 matrix as given below.

$$\begin{bmatrix} Y_1 \\ Y_2 \end{bmatrix} = \begin{bmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$$

Alternatively, the input-output relationship can also be characterized by a chain matrix

$$\begin{bmatrix} X_1 \\ Y_1 \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} Y_2 \\ X_2 \end{bmatrix}$$

For a general n-port digital filter structure, structural boundedness is equivalent to saying that the matrix T(z) is <u>contractive</u> [16] i.e.*

$$T^{t*}(z^{-1})T(z)\Big|_{z=e^{j\omega}} - I \leq 0 \quad \forall\, \omega \tag{2.3}$$

where superscript 't' denotes transposition.

If equality sign holds in eqn. (2.3) for all ω then T(z) is the transfer matrix of a <u>lossless filter</u>, i.e., input and output inner products are equal. In such a case

$$Y^{t*}Y = X^{t*}X\text{ ,} \quad \text{where t denotes transposition}$$

or,

$$\sum_i |y_i(n)|^2 = \sum_i |x_i(n)|^2$$

*: In other words, $T^{t*}(Z^{-1})\ T(Z)|_{Z=e^{j\omega}}$ is negative semi-defini

Thus the lossless nature of the transfer matrix can be looked upon as the generalised notion of lossless boundedness. We will see in a later section how this can provide a low sensitivity and stable design for digital filters for parallel processing of signals.

SECTION THREE

WAVE DIGITAL FILTERS : INDIRECT REALISATION OF LOW SENSITIVITY FILTERS

3.1 INTRODUCTION

We are by now aquainted with a class of analog filters viz. doubly terminated lossless filters. It was shown in the previous section that a lossless network inserted between two resistive terminations possessed low sensitivity with respect to its element-values. The property was seen to be the result of the structural passivity of the filter network and the restrictions placed upon it by the maximum power transfer theorem. By analogy, it was later seen that the low-sensitivity also extended to digital filters whose filter functions were structurally bounded.

In this section, we discuss an important class of digital filters that imitate analog doubly terminated lossless filters in the digital domain. These filters are known as 'Wave digital filters' (WDFs) and were first introduced by A. Fettweis [1]. Owing to their popularity, now they have also been included in some text-books [17]. A brief overview and details of the current state of art on WDFs can be found in a recent tutorial paper [18].

In the discussions, we cover the basic concepts and design techniques of these filters. Their low sensitivity property has also been explained. Some simulation results have been included to give a more concrete picture of WDFs to the reader (Appendix B)

3.2 PRELIMINARIES

Our purpose is to design digital filters that exhibit very low sensitivity to changes in multipliers. An attractive approach is to consider low sensitivity analog filters and to try to find analogous structures in the digital domain. If a suitable transformation can be found, then we can realise low sensitivity digital filters by an indirect method. This would also obviate the need to develop entirely new design techniques in digital domain. WDFs, however are still very attractive because of their simple design techniques.

Bilinear Transformation

To synthesize wave digital filters, a transformation is required by which an analog network is converted into an equivalent digital structure. The transformation should be such that none of the stability conditions are violated. The design of WDFs is carried out in the transformed domain, denoted by ψ. This domain is analogous to the actual complex frequency s-domain, which is henceforth called the reference

domain. The relationship between ψ, s and z is given by

$$z = e^{sT} = \frac{1+\psi}{1-\psi} \qquad (3.1a)$$

or,

$$\psi = \frac{1-z^{-1}}{1+z^{-1}} = \mathrm{Tanh}\,(sT/2) \qquad (3.1b)$$

Eqn. (3.1b) is similar to one used in the design of microwave filters to convert exponentials into rationals. Here T denotes a unit of delay, most often the Nyquist sampling period. Fig. (3.1) shows that the area inside unit circle in the z-plane is mapped onto LHP in the ψ-plane. Therefore, stability conditions in the z-domain are preserved since all poles inside the unit circle z-plane are transformed to LHP poles in the reference domain.

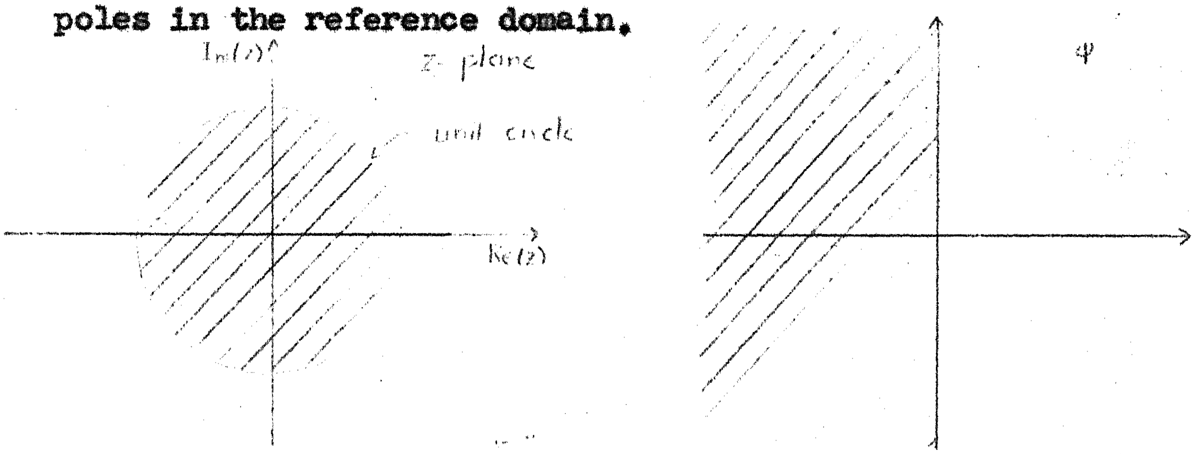


Fig. 3.1 Mapping of z-plane on ψ-plane.

Signal representation

In the reference domain, all ports are characterized by relationships between corresponding voltage (V) and current (I)

signals. In the case of wave digital filters, these signals are represented by the so called 'wave quantities'. These are the incident and reflected wave signals, denoted by A and B respectively. Also, each port is associated with a port resistance R which can be arbitrarily chosen. Relationships between these signals are as follows

$$A = V + RI, \quad B = V - RI \qquad (3.2a,b)$$

Representation of elements

For the design of WDFs, elements viz. resistors, capacitors, inductors and voltage sources are treated as single ports. Unit elements are, however 2-ports. Using eqns. (3.2), all these elements can be transformed into equivalent digital sub-structures as shown in Fig. (3.2). For illustration of this, an inductor is considered here.

EXAMPLE 3.1

Find the equivalent WDF representation of an inductor.

For an inductor, voltage-current relationship in ψ-domain is given by

$V = (\psi L)I$, L being the value of inductance.

If we choose port resistance $R = L$, then eqn. (3.2) gives

$$A = (1 + \psi)\,RI\,, \quad B = -(1 - \psi)\,RI$$

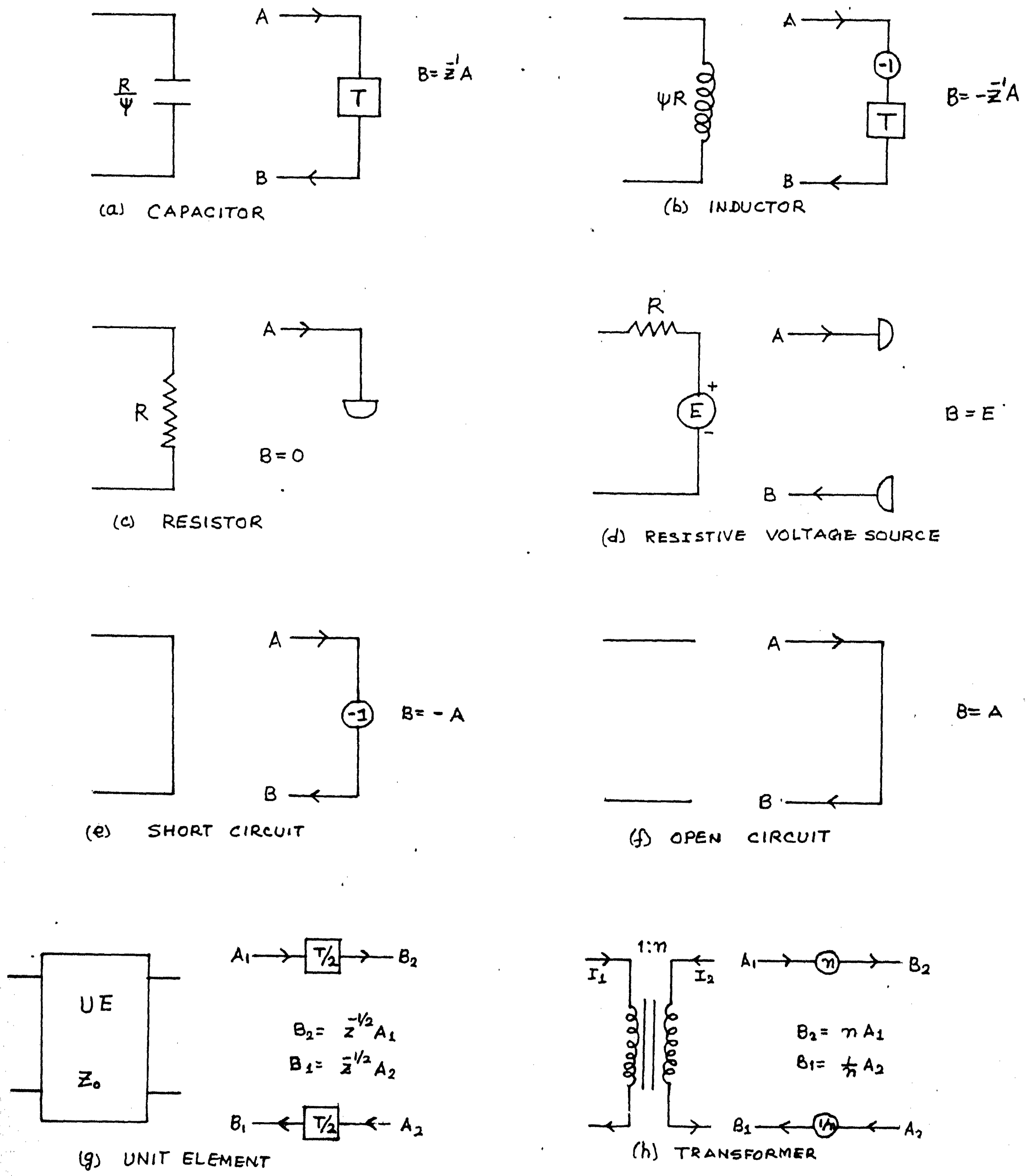


Fig. 3.2 Major elements and their realisation in WDF domain.

Hence $$B = -\frac{(1-\psi)}{(1+\psi)} A$$

and eqn. (3.1a) gives $B = -z^{-1}A.$

Thus an inductor in the ψ-domain is equivalently represented by a delay in the z-domain as shown in Fig. (3.3).

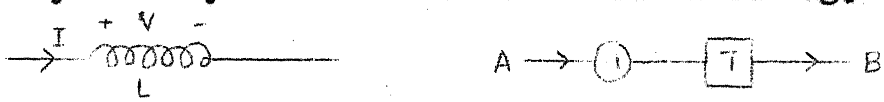


Fig. 3.3 Inductor and its WDF equivalent of Example (3.1)

Interconnection of Ports

We have seen how it is possible to simulate in the digital domain the main elements of the reference filter. To complete the equivalence, we also need to satisfy various topological rules such as the Kirchoff's laws in the reference domain. Similarly, various ports in wave digital filters can be connected either in series on in parallel. However, unless port resistances are identical, these ports cannot be connected directly without violating eqn. (3.2). To overcome this difficulty, a multiport element called an '<u>adaptor</u>' is used for such interconnections. Adaptors play an important role in the realisation of WDFs as it will be seen in the following discussion.

3.3 ADAPTORS

As stated earlier, adaptors are used to connect digital ports either in series or in parallel [19]. They are

basically 'standardized' digital structures that consist of only multipliers and adders. There can be both serial and parallel adaptors depending upon their functions. These standardized n-port adaptors are shown in Fig. (3.4). To illustrate how a typical adaptor equations are obtained, we consider the following example.

EXAMPLE 3.2

Design a parallel adaptor for interconnecting two ports with port resistances R_1 and R_2 respectively.

Eqns. (3.2) gives the general description as

$$A_1 = V_1 + I_1R_1, \qquad B_1 = V_1 - I_1R_1$$

$$A_2 = V_2 + I_2R_2, \qquad B_2 = V_2 - I_2R_2$$

For parallel interconnection of ports, we have

$$V_1 = V_2 \quad \text{and} \quad I_1 = -I_2$$

or,

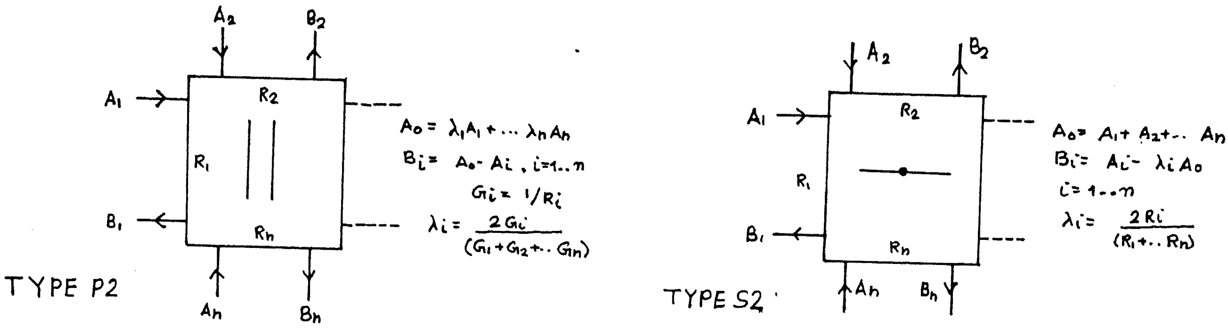
$$A_1 + B_1 = A_2 + B_2 \tag{3.3a}$$

and,

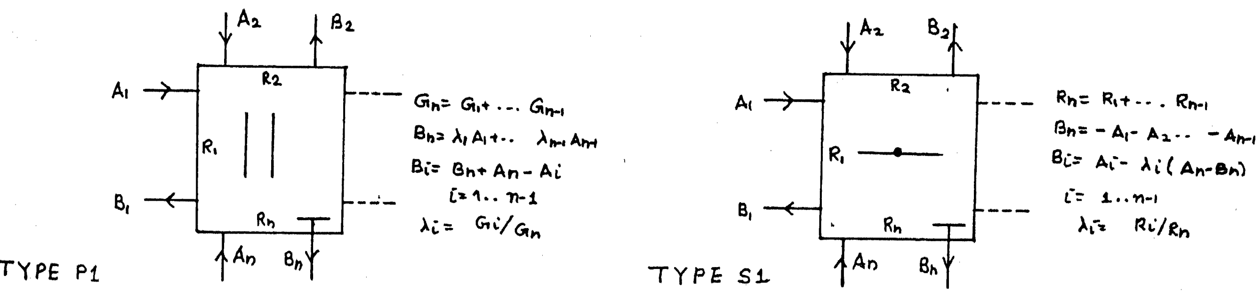
$$\frac{A_1 - B_1}{R_1} + \frac{A_2 - B_2}{R_2} = 0 \tag{3.3b}$$

Solving eqn. (3.3) for B_1 and B_2, we get

$$B_2 = \lambda_1 A_1 + (1 - \lambda_1) A_2$$



(a) Unconstrained parallel and series adaptors



(b) Constrained parallel and series adaptors

Fig. 3.4 Standardized N-port adaptors.

$$B_1 = B_2 + A_2 - A_1, \quad \text{where } \lambda_1 = \frac{2G_1}{(G_1 + G_2)}$$

$$G_1 = 1/R_1; \; G_2 = 1/R_2$$

the 2-port adaptor is shown in Fig. (3.5).

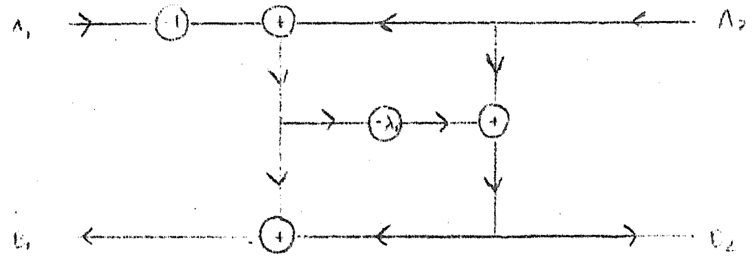


Fig. 3.5 Two-port parallel adaptor of Example (3.2).

Serial and parallel adaptors, named P2 and S2 respectively in Fig. (3.4) are <u>unconstrained</u>. The name is given because there is no restriction on the values of the port resistances. These adaptors require n-1 multipliers, each λ_i (i = 1 to n-1) being an independent quantity. Equations in Fig. (3.4b), however, suggest that if λ_i is unity for a port i, then its corresponding reflected signal B_i is independent of the incident signal A_i. Thus the port becomes '<u>reflection free</u>'. An adaptor having a reflection free port is called a <u>constrained adaptor</u> and is denoted by P! or S!. A constrained adaptor requires only n-2 multipliers for its realization.

A reflection free port can always be connected to any other port without resulting in a delay free loop. Such

loops should be absent to have a computable digital structure. Since there is no path from the input signal to the output signal of a reflection free port, a delay free loops in this case are automatically avoided. If certain precautions (A.7) are kept in mind, then computable WDF structures can be obtained.

3.4 TRANSFER FUNCTION REALISATION

Having considered various building blocks of a WDF structure, we now consider how a transfer function denoted by $H_A(\psi)$ in the reference domain is realised in the digital domain. Similar to the analog case, WDF realisation is based on the design of a 'psuedolossless' network terminated by an 'equivalent' resistive element (open circuit in the WDF design).

Consider again the network of Fig. (1.1). The input-output port equations can be written as

$$A_1 = V_1 + I_1R_1 , \qquad B_1 = V_1 - I_1R_1$$

$$A_2 = V_2 + I_2R_2 , \qquad B_2 = V_2 - I_2R_2$$

Since the output port is resistively terminated, we have

$$V_2 = - I_2R_2 \quad \text{or} \quad A_2 = 0$$

Further,

$$B_2 = 2V_2, \quad A_1 = V_1 + I_1R_1 = V_s \tag{3.4}$$

If H(z) denotes the transfer function in digital domain then eqn. (3.4) gives

$$H(z) = \left.\frac{B_2}{A_1}\right|_{A_2=0} = 2\frac{V_2}{V_s} = 2H_A(\psi)\Big|_{\psi=(z-1)/(z+1)}$$

The 2-port digital structure is shown in Fig. (3.6) which realizes the transfer function H(z).

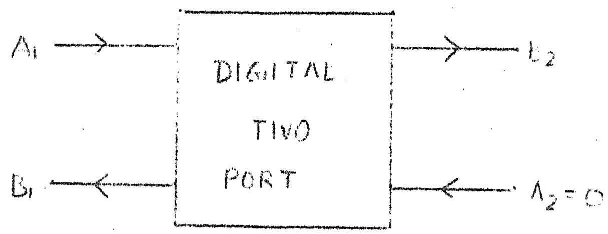


Fig. 3.6 A two-port digital structure.

3.5 LOW SENSITIVITY OF WDF's

That wave digital filters show low sensitivity to the coefficients' quantization can be intuitively understood. This is so because WDFs are obtained by a bilinear transformation of the reference filters having this property. However, to explain the low sensitivity of WDFs more rigorously, the concepts of 'pseudolosslessness' and 'pseudopower' need be defined for wave quantities in the digital domain [2].

Consider for a 2-port, the total psuedopower P, absorbed by it which is defined as

$$P = \frac{|a|^2 - |b|^2}{R} \tag{3.5}$$

where R is the port resistance and a and b denote input and output wave quantities respectively. In the case of

doubly terminated filters, it was seen in Section II that the inserted network consisted of lossless elements alone. This resulted in MAP frequencies in passband with zero sensitivity at these points. On similar grounds it can be shown by an identical reasoning that if the elements in WDF structures are pseudolossless then WDFs have the low sensitivity property. We consider as an illustration, the case of a 2-port series adaptor.

EXAMPLE 3.3

Show that a 2-port series adaptor, as shown in Fig. (3.7), is pseudolossless, i.e. it does not absorb any pseudopower.

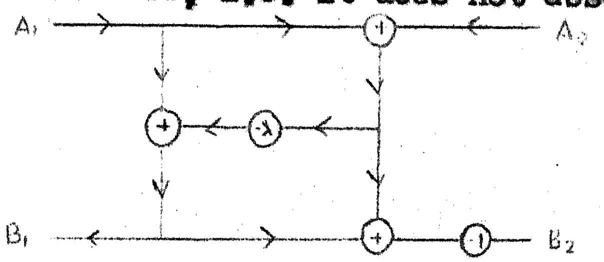


Fig. 3.7 Two-port series adaptor of Example (3.3).

The port equations are given by

$$B_1 = (1 - \lambda)\, A_1 - \lambda A_2$$

$$B_2 = -(2-\lambda)\, A_1 - (1 - \lambda) A_2 \qquad (3.6)$$

Total power absorbed by the adaptor is

$$P = \frac{A_1^2 - B_1^2}{R_1} + \frac{A_2^2 - B_2^2}{R_2}$$

$$= \frac{(A_1 - B_1)(A_1 + B_1)}{R_1} + \frac{(A_2 - B_2)(A_2 + B_2)}{R_2} \qquad (3.7)$$

Substituting eqn. (3.6) in (3.7) and simplifying, we get

$$P = ((2-\lambda)A_1 - \lambda A_2)(A_1+A_2)(\frac{\lambda}{R_1} - \frac{2-\lambda}{R_2})$$

Substituting for λ in the last term, we find that $P = 0$. Thus the adaptor is a psuedolossless element.

Similarly, other elements including delays too can be shown to be pseudolossless. A resistor, i.e. an open circuit however, is seen to be a passive element. Therefore, all results of the analog case apply here as well. Hence the low sensitivity nature of wave digital filters is seen to be preserved.

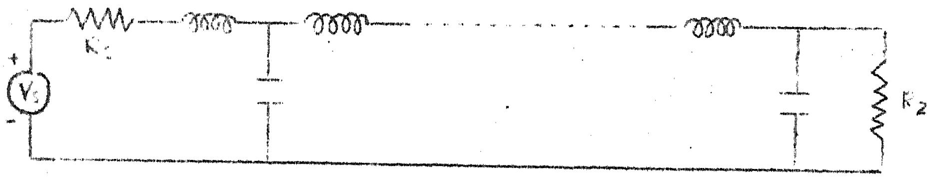
3.6 SYNTHESIS OF WAVE DIGITAL FILTERS

We now consider various kinds of WDF structures that are based on the synthesis of analogous reference domain filters. These filters were discussed in Section I and were later shown to possess the low sensitivity property. Namely, we can design ladder or lattice wave digital filters. Besides these, unit element based WDF structures are derived from the corresponding microwave filters.

Ladder wave digital filters

Ladders are the most common structures for wave digital filters. In these structures, serial and parallel adaptors alternate, imitating the series and shunt connections of the reference filter as shown in Fig. (3.8). Each adaptor

is a 3-port, the second port representing either a capacitor or an inductor. In WDF, an L or C element involves a delay. Thus the structure is canonic in delay elements. The 3rd port resistance is so chosen (A.8) that each adaptor except the last one is a constrained adaptor.[20]. This makes the design of adaptors possible by using only one multiplier each. Since the last adaptor is unconstrained, total number of multipliers needed are n+1. To illustrate, we consider an example of ladder WDF structure.



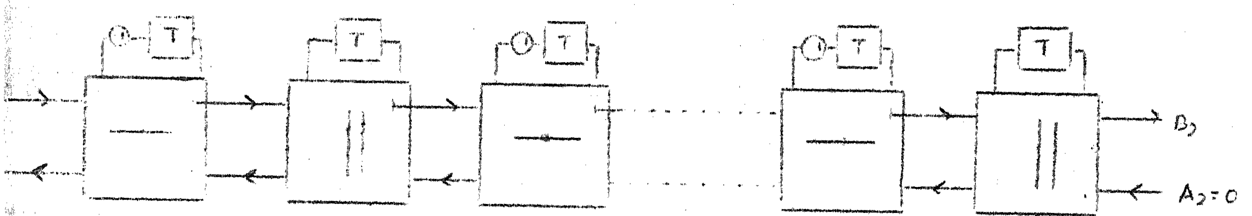


Fig. 3.8 A ladder based WDF structure.

EXAMPLE 3.4

The filter network of Fig. (1.2) can be redrawn as Fig. (3.9). Derive its equivalent WDF ladder structure.

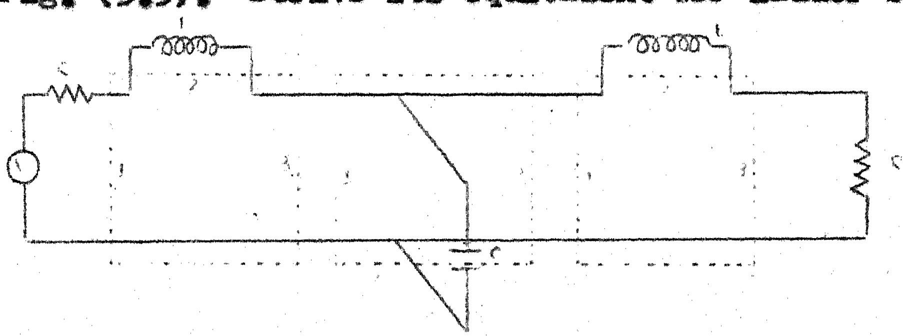


Fig. 3.9 Third order Chebyshev filter of Example (3.4).

The equivalent WDF structure is easily drawn as in Fig. (3.10).

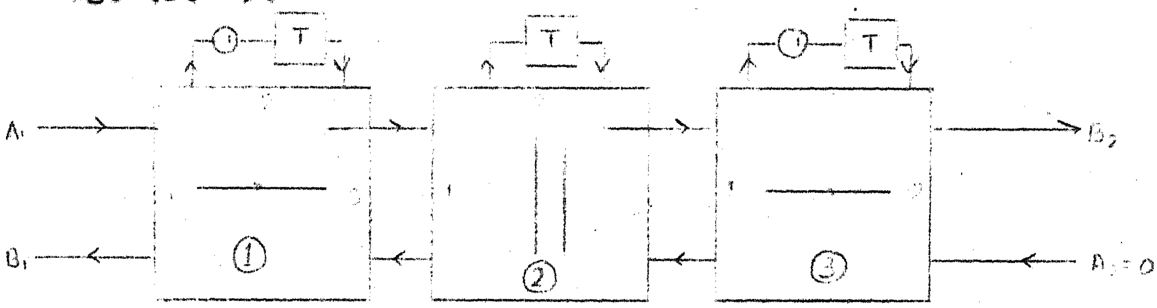


Fig. 3.10 Ladder based WDF structure of Example (3.4).

Choosing port 2 as the dependent port for each adaptor, we have

Adaptor 1 : type S1

$R_1 = 1$, $R_2 = 2$, $R_3 = R_1+R_2 = 3$

$\lambda_{11} = 1/3$

Adaptor 2 : type P1

$G_1 = 1/3$, $G_2 = 1$, $G_3 = G_1+G_2 = 4/3$

$\lambda_{21} = 1/4$

Adaptor 3 : type S2

$R_1 = 3/4$, $R_2 = 1$, $R_3 = 2$

$\lambda_{31} = 2/5$, $\lambda_{32} = 3/15$

It is observed that the third adaptor requires 2 multipliers because it is unconstrained as none of its port resistances can be arbitrarily chosen.

Unit element based structures

An attractive structure for wave digital filters is obtained by the corresponding microwave filter design [21]. Since unit elements in wave digital filters are realised by half delays, these delay based structures have several advantages. Firstly, the delay-free loops which may be generated in other designs are very easily avoided. Secondly, the use of delay elements reduces the number of multipliers in each delay-free path. This saves the computation time required for the processing of each input data.

In the following example we consider design of a unit element based WDF.

EXAMPLE 3.5

Design a unit element based WDF from the reference filter of Fig. (1.10).

Referring to Fig. (1.10), its equivalent WDF structure is drawn in Fig. (3.11).

In this case, it is observed that all adaptors are unconstrained (type P2). Specifications of these are

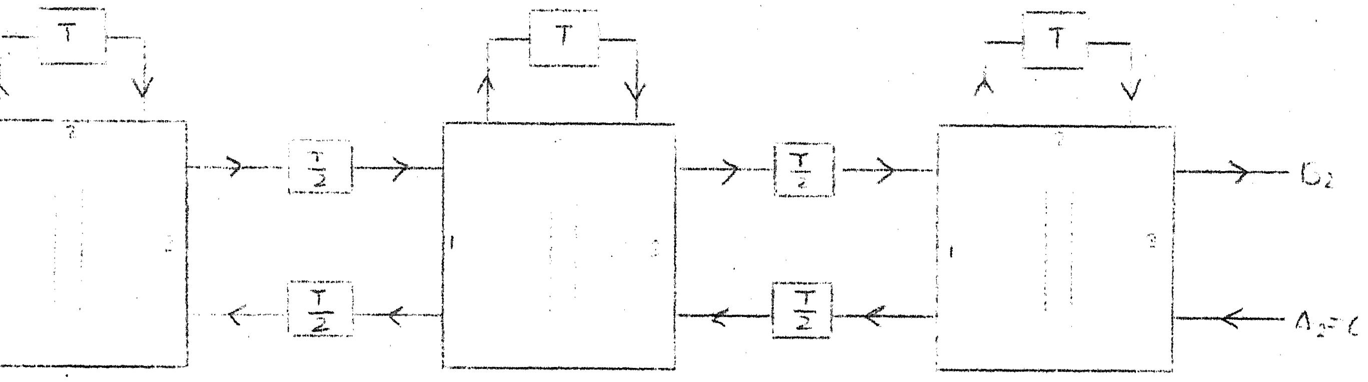


Fig. 3.11 Unit-element based WDF structure of Example (3.5).

Adaptor 1 :

$$G_1 = 1, \quad G_2 = 2, \quad G_3 = 3/4$$

$$\lambda_{11} = 8/15, \quad \lambda_{12} = 2/5$$

Adaptor 2 :

$$G_1 = 3/4, \quad G_2 = 9/4, \quad G_3 = 3/5$$

$$\lambda_{21} = 5/12, \quad \lambda_{22} = 1/3$$

Adaptor 3 :

$$G_1 = 3/5, \quad G_2 = 2/3, \quad G_3 = 1$$

$$\lambda_{31} = 9/17, \quad \lambda_{32} = 10/17.$$

A point to note, as shown in Fig. (3.12) is that any two half delays preceding a digital port can be joined together to form a full delay element. Thus in actual

design, no half delays are used. The digital structure of Fig. (3.11) can, therefore, be redrawn as Fig. (3.13).

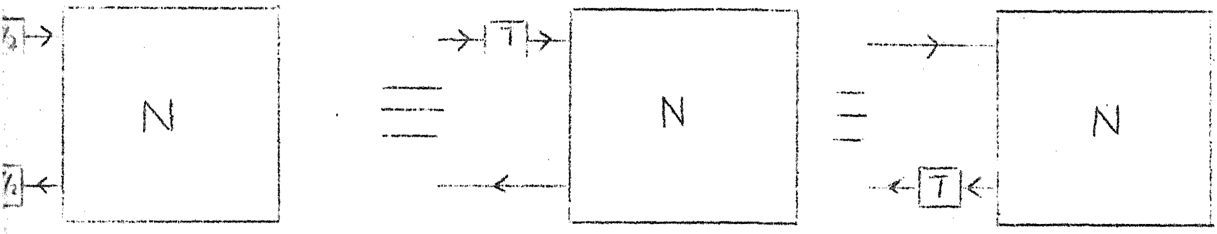


Fig. 3.12 Equivalence of two half-delays and a full-delay element.

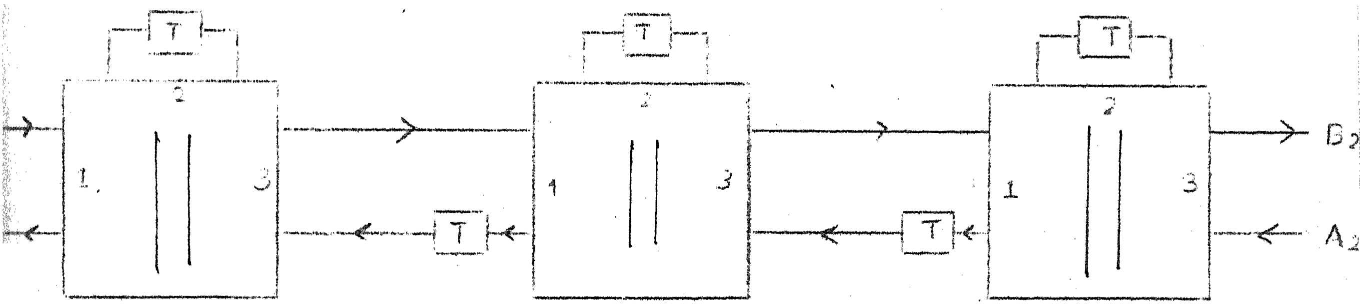


Fig. 3.13 Delay based WDF structure of Example (3.5).

Lattice Wave Digital Filters

Lattice wave digital filters are also an important class of wave digital filters. Obtained from the lattice reference filters in similar fashion as others, they become more attractive because symmetry of the network requires

fewer multipliers in the digital domain [22].

Consider the lattice network of Fig. (3.14).

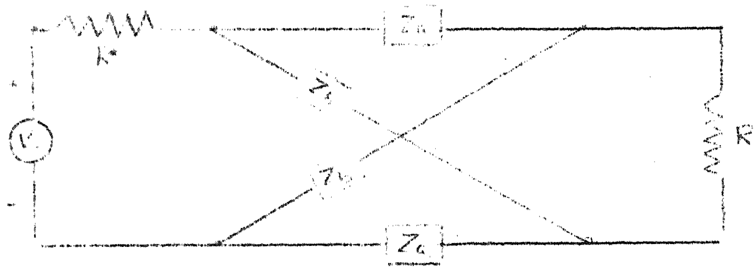


Fig. 3.14 A symmetric lattice network.

Here both Z_a and Z_b are lossless reactance functions. Just as we had used earlier an impedance transfer matrix [Z] for the analog case, here we define a scattering matrix [S] which relates input and output wave quantities by

$$\underline{B} = [S]\,\underline{A}$$

where,

$$\underline{B} = [B_1\ B_2]^t,\quad \underline{A} = [A_1\ A_2]^t \qquad (3.8)$$

and

$$[S] = \begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix}$$

It is observed (A.9) that for a symmetric case

$$s_{11} = s_{22} = \tfrac{1}{2}(s_a + s_b)$$

$$s_{21} = s_{12} = \tfrac{1}{2}(s_b - s_a)$$

where, $\quad s_a = \dfrac{Z_a - R}{Z_a + R}, \quad s_b = \dfrac{Z_b - R}{Z_b + R}.$

Hence, the transfer function is given by

$$H(z) = \frac{B_2}{A_1}\bigg|_{A_2=0} = s_{21} = \frac{1}{2}(s_b - s_a)$$

The transfer function is realised as in Fig. (3.15).

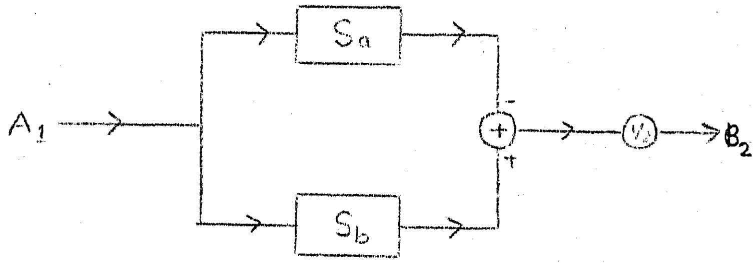


Fig. 3.15 A lattice realisation of wave digital filter

Both s_a and s_b are obtained by the transformation of reactance functions. Consider an example for illustration.

EXAMPLE 3.6

Design a lattice wave digital filter for the specifications as in example (1.2).

In this case, Z_a and Z_b are obtained as

$$Z_a = 2s, \qquad Z_b = 2s + 2/s$$

After transformation, they are redrawn in Fig. (3.16).

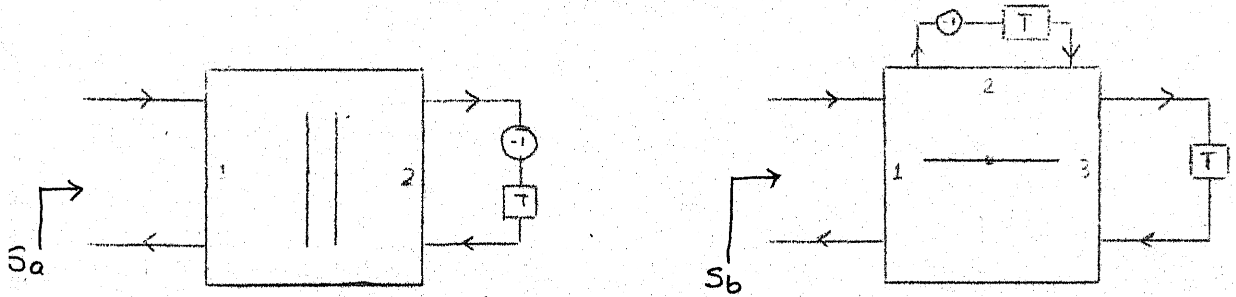


Fig. 3.16 Lattice WDF structure of Example (3.6).

Adaptors in Fig. (3.16) have the following parameters

Adaptor 1 : type P1

$G_1 = 1$, $G_2 = 1/2$ and $\lambda_{12} = 2/3$

Adaptor 2 : type S2

$R_1 = 1$, $R_2 = 1/2$, $R_3 = 2$

$\lambda_{21} = 4/7$, $\lambda_{22} = 2/7$.

As a variation, one can also apply Richard's theorem for Realizing Z_a and Z_b which are lossless functions. This gives a general method of designing lattice WDFs by using only the unit elements.

State Space WDF Structures

WDF structures discussed so far were based on direct implementation of transformed structures in the digital domain. State space were digital filters, however, are implemented using the state variables [23].

In general state space digital filters are expressed by the following difference equations.

$$\underline{X}(n+1) = [A]\,\underline{X}(n) + [B]\,\underline{U}(n)$$

$$\underline{Y}(n) = [C]\,\underline{X}(n) + [D]\,\underline{U}(n) \qquad (3.9)$$

where $\underline{X}$, $\underline{Y}$ and $\underline{U}$ are the state, output and input vectors respectively.

A WDF structure derived from the reference filters by any of the earlier methods can be looked upon as an (N+2) - digital multiport. Here N is the total number of delay elements corresponding to the inductors and capacitors. If these delays along with the input and output ports are taken out as shown in Fig. (3.17) then the entire network can be considered a generalised (N+2)-port adaptor.

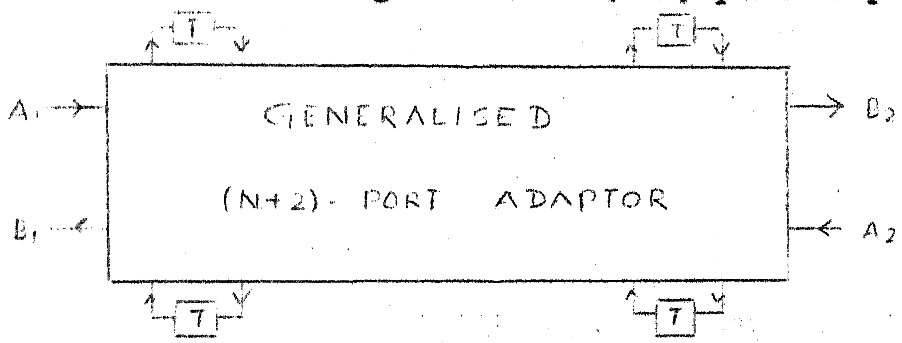


Fig. 3.17 Generalised (N+2)-port adaptor for WDF.

For a port i, terminated in a delay element, the corresponding equation is

$$A_i = z^{-1} B_i$$

or,

$$A_i(n+1) = B_i(n) \tag{3.10}$$

If we choose A_i's (i = 1 to N) as N state variables, A_1 as input and B_2 as output, then the entire WDF structure can be implemented as a state space model. This is illustrated by the following example.

EXAMPLE 3.7

Realise the WDF of Fig. (1.2) as a state space digital filter (SSDF).

The WDF to be modelled as a SSDF is redrawn in Fig. (3.18).

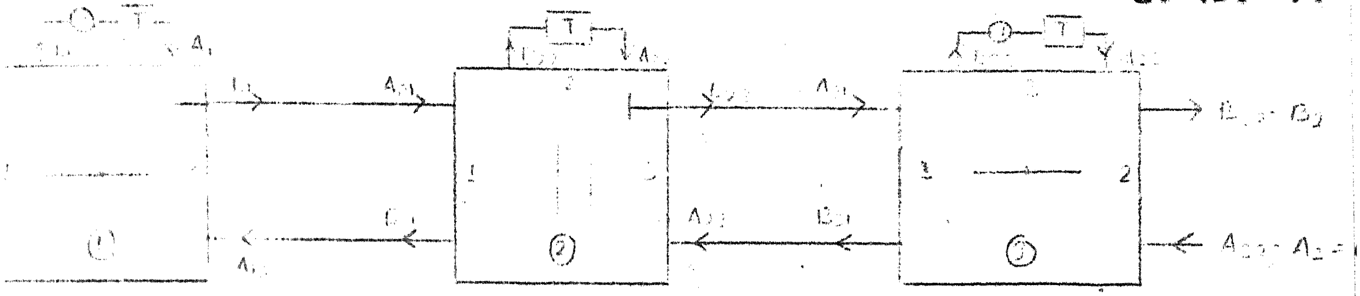


Fig. 3.18 WDF structure for SSDF realisation of Example (3.7).

If input and output wave quantities are denoted by A_{ij} and B_{ij} respectively, where

i = adaptor number

j = port number.

Then from Fig. (3.18) it is seen that,

$$B_{13} = A_{21}, \quad A_{13} = B_{21}, \quad A_{23} = B_{31}, \quad B_{23} = A_{31}$$

Replacing A_{12} and A_{33} by $-A_{12}$ and $-A_{33}$ respectively to eliminate the negative sign before the delays, we get

$$B_{12} = -\frac{16}{15}A_{11} + \frac{1}{15}A_{12} - \frac{4}{5}A_{22} - \frac{4}{15}A_{33}$$

$$B_{22} = \frac{1}{5}A_{22} - \frac{2}{3}A_{11} + \frac{2}{5}A_{12} + \frac{2}{5}A_{33}$$

$$B_{33} = -\frac{4}{5}A_{22} + \frac{4}{15}A_{11} - \frac{4}{15}A_{12} + \frac{1}{15}A_{33}$$

$$B_{32} = \frac{8}{15}A_{33} - \frac{2}{5}A_{22} + \frac{2}{15}A_{11} - \frac{2}{15}A_{12} \qquad (3.11)$$

Using U for A_{11}, $[X_1\ X_2\ X_3]^t$ for $[A_{11}\ A_{22}\ A_{33}]^t$ and Y for B_{32}, we write the state equations from eqns. (3.9) and (3.11) as

$$\begin{bmatrix} X_1(n+1) \\ X_2(n+1) \\ X_3(n+1) \end{bmatrix} = \begin{bmatrix} 1/15 & -4/5 & -4/15 \\ 2/5 & 1/5 & 2/5 \\ -4/15 & -4/5 & 1/15 \end{bmatrix} \begin{bmatrix} X_1(n) \\ X_2(n) \\ X_3(n) \end{bmatrix} + \begin{bmatrix} -16/15 \\ -2/5 \\ 4/15 \end{bmatrix} U(n)$$

$$Y(n) = \left[-\frac{2}{15}\ \ -\frac{2}{5}\ \ \frac{8}{15}\right] \begin{bmatrix} X_1(n) \\ X_2(n) \\ X_3(n) \end{bmatrix} + \left[\frac{2}{15}\right] U(n) \qquad (3.12)$$

Which is the desired state space formulation.

In this section, we focussed our attention on an important class of low sensitivity digital filters viz. wave digital filters. It was shown how a large number of possible designs in the reference domain are also useful for the design of digital filter structures using wave quantities. The method, although indirect, also gives good performance with respect to the quantization of multipliers. In the next section, we will consider some direct approaches to the design of low sensitivity digital filters that do not require a prior knowledge of analog filters.

SECTION FOUR

DIRECT REALISATION OF LOW SENSITIVITY DIGITAL FILTERS

4.1 INTRODUCTION

In the previous section, we discussed the theory and design techniques of wave digital filters. The synthesis of these filters requires a knowledge of analog filter theory as well. The method of realisation is an indirect one because any z-domain transfer function, H(z), is first transformed into its corresponding transfer function in the reference domain. It is then realised in the reference domain and transformed back into the z-domain. It may sometimes happen that no low sensitivity reference filter exists for a specified H(z). Thus the indirect method is not universally applicable. Besides this, the choice of digital filter structures is limited by the number of different reference filter networks for a given transfer function. Further, in this approach the optimization of multipliers, adders and delays etc. is carried out only for the building blocks viz. adaptors etc., and not for the entire filter structure as a whole. The method, therefore, gives only a suboptimal realisation for low sensitivity digital filter characteristics.

In this section, we examine some of the direct realisation methods for low sensitivity digital filters. We

start with a ladder/lattice structure proposed by Gray and Markel [3]. We consider the concept of 'embedding' of a bounded function into a scattering matrix for its 2-port structure realisation. We also discuss how this matrix can be synthesized by the so called LBR-extraction approach [6]. Later in the discussions, we consider the realisation of these filters by cascading of 'orthagonal' blocks that constitute what are known as orthogonal digital filters [5]. Finally we investigate the common features of all of these methods and then try to establish a link between them by presenting a unified approach for the design of low sensitivity digital filters in the next section.

4.2 LATTICE OR LADDER BASED DIGITAL FILTERS

Direct realisation of low sensitivity digital filters was first attempted by Gray and Markel [3]. An algorithm proposed by them gives a cascaded digital filter structure. Each building block can be realised either as a 'ladder' or as a 'lattice' structure in the digital domain. To illustrate this, consider a transfer function H(z) given by

$$H(z) = \frac{P_M(Z)}{A_M(Z)} \tag{4.1}$$

where, $P_M \; := P_M(Z) = \sum_{n=0}^{m} p_{m,n} \, z^{-n}, \; m = 1,2,\ldots M$

$$A_M := A_M(z) = 1 + \sum_{n=1}^{m} a_{m,n} z^{-n}$$

This transfer function is directly realisable by using 2M+1 multiplier coefficients but this does not yield a low sensitivity structure. As we shall presently see, GRAY and MARKEL's algorithm also requires 2M+1 parameters and is, in that sense, canonical. Let us define intermediate polynomials $A_m(z)$ and $B_m(z)$ of the forms given in (4.1) for $m = 0,1,2,\ldots M$. Then the algorithm (A.10) gives the following realisation of H(z).

$$H(z) = \sum_{m=0}^{M} v_m \frac{zB_m(z)}{A_M(z)} \qquad (4.2)$$

Here v_m are called the <u>tap parameters</u>. $A_m(z)$ and $B_m(z)$ at any stage m are related (A.11) as

$$\begin{bmatrix} A_{m+1} \\ B_{m+1} \end{bmatrix} = \begin{bmatrix} 1 & k_m \\ z^{-1}k_m & z^{-1} \end{bmatrix} \begin{bmatrix} A_m \\ B_m \end{bmatrix} \qquad (4.3)$$

k_m are called <u>k-parameters</u>. The algorithm (A.10) gives the values of tap parameters and k-parameters for any bounded transfer function. If X(z) and Y(z) denote the input and output sequences respectively then Y(z) can be written as

$$Y(z) = G(z)\,X(z) = \sum_{m=0}^{M} v_m \frac{zB_m(z)}{A_M(z)} X(z) \qquad (4.4)$$

Eqn. (4.3) can be realised as a 2-port <u>lattice structure</u> which is shown in Fig. (4.1).

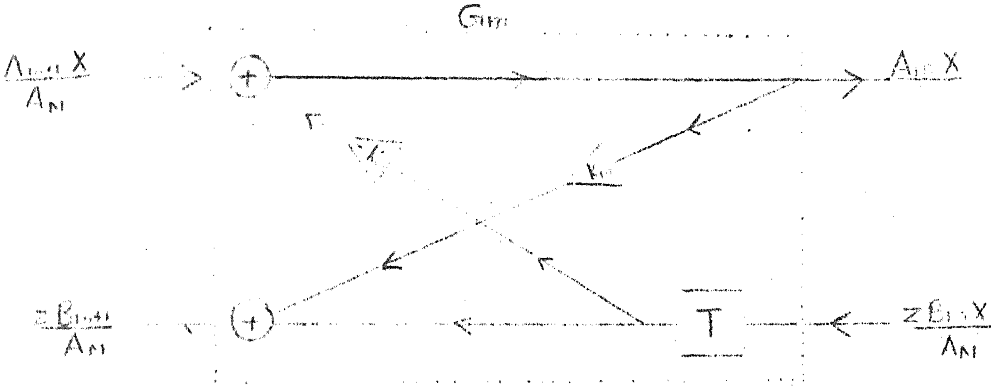


Fig. 4.1 Two-port lattice structure.

Using this lattice structure and eqn. (4.2), we can realise the cascaded form of digital filter structure as shown in Fig. (4.2).

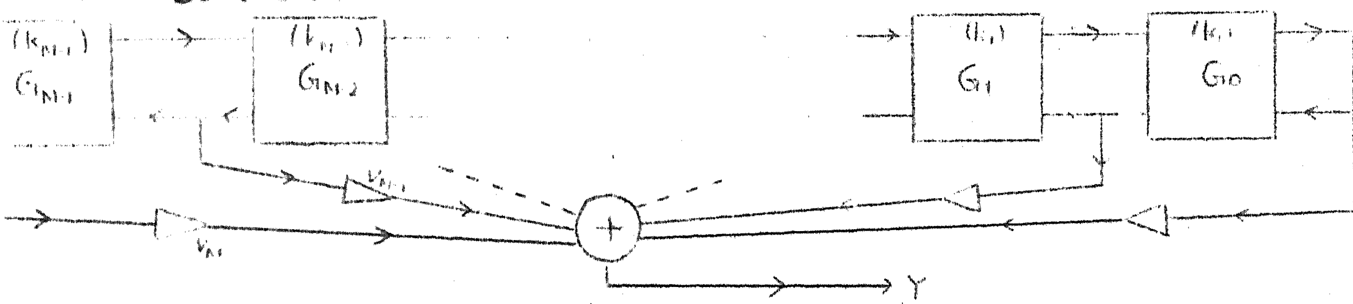


Fig. 4.2 Cascaded digital filter structure of Gray and Markel.

Alternatively, a rearrangement of terms leads to a <u>ladder structure</u> (figure (4.3)). The transfer matrix of a 2-port ladder is given as

$$\begin{vmatrix} A_m \\ zB_{m+1} \end{vmatrix} = \begin{bmatrix} 1 & -k_m \\ k_m & 1-k_m^2 \end{bmatrix} \begin{bmatrix} A_{m+1} \\ B_m \end{bmatrix} \qquad (4.5)$$

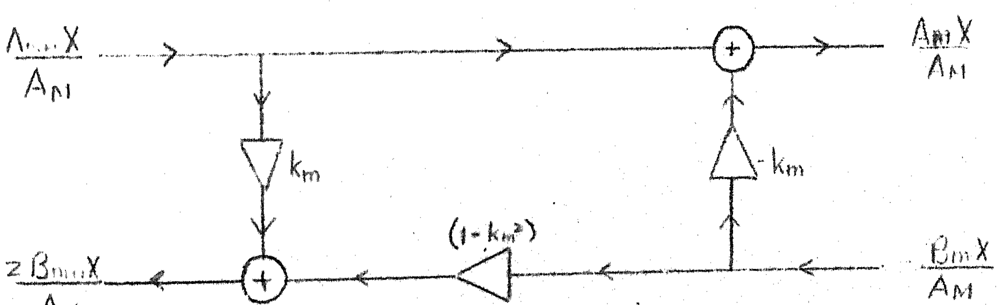


Fig. 4.3 Two-port Ladder structure.

By using the concepts of 'pseudopower', similar to that in case of the wave digital filters, it can be shown that the GRAY and MARKEL structure is also structurally passive and therefore, shows low sensitivity to changes in coefficients due to quantization [24]. It will be illustrated later in this section that this is due to the 'orthogonality' of the transfer matrix of the ladder or lattice building blocks.

Let us now consider an example of this algorithm.

EXAMPLE 4.1

Design a Gray and Markel lattice filter with following transfer function

$$H(z) = \frac{0.4(1 + 2z + z^{-2})}{(1 + 0.89\, z^{-1} + 0.16\, z^{-2}}$$

Here,

$$P_2 = .4 + .8\, z^{-1} + .4\, z^{-2}$$

$$A_2 = 1 + .89\, z^{-1} + .16\, z^{-2}$$

From steps (1.2, 1.3 and 1.4) the algorithmgives

$$zB_2 = 0.16 + 0.89\, z^{-1} + z^{-2}$$

$$k_1 = a_{22} = 0.16$$

$$v_2 = p_{22} = 0.4$$

Steps (1.5) and (1.6) give

$$A_1 = (A_2 - k_1 zB_2)/(1 - k_1^2)$$

$$= 1 + 0.7672413\ z^{-1}$$

$$P_1 = P_2 - v_2\ zB_2$$

$$= 0.336 + 0.444\ z^{-1}$$

Hence,

$$zB_1 = 0.7672413 + z^{-1}$$

repeating the process, we obtain

$$k_o = 0.7672413$$

$$v_1 = 0.444$$

$$v_o = 1.0$$

the lattice structure is implemented as in Fig. (4.4).

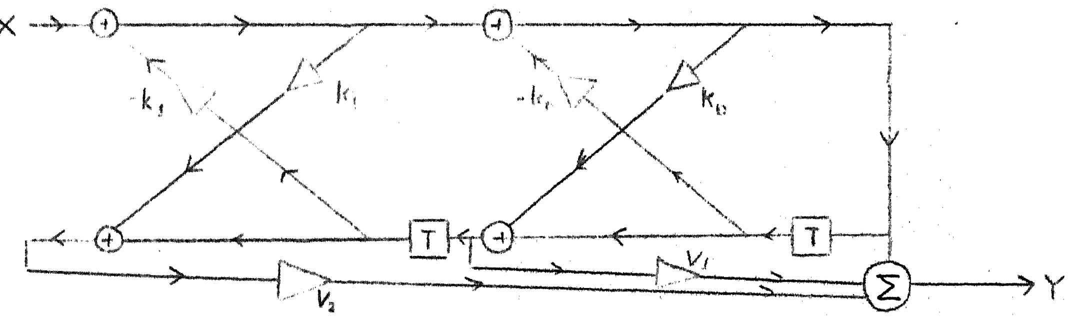


Fig. 4.4 Lattice filter structure of Example (4.1).

4.3 TRANSFER FUNCTION EMBEDDING

An alternative method to realise a transfer function is to 'embed' it in a digital 2-port transfer matrix. It

was illustrated in section II how a lossless bounded function is insensitive to parameter perturbations. In this case, low sensitivity is 'enforced'. On similar grounds, we can intuitively argue that a lossless matrix too, which is a generalized notion of lossless bounded functions, shows low sensitivity to its parameters.

Evidently, to achieve this, it should be possible to embed a <u>lossy transfer function</u>, i.e. one which is not lossless bounded, into a lossless matrix. If H(z) is any causal and strictly stable rational transfer function which is bounded above by unity, then it can always be embedded into a matrix T(z) which is lossless. This result is based on the classical embedding procedure given by Belevitch for scattering functions into a para unitary matrix [25].

If the transfer function H(z) is given by $H(z) = f(z)/g(z)$, and is embedded as T_{21} (figure 4.5a) in the matrix T(z), then the T(z) is given by

$$T(z) = \frac{1}{g(z)} \begin{bmatrix} -h_*(z) & f_*(z) \\ f(z) & h(z) \end{bmatrix} \qquad (4.6)$$

where $f_*(z) := z^{-N} f(z^{-1})$, $h_*(z) := z^{-N} h(z^{-1})$

N is degree of the polynomial g(z).

For T(z) to be lossless, the following relationship holds between f, g and h

$$ff_* + hh_* = gg_* \qquad (4.7)$$

From the knowledge of f and g, one can obtain h using eqn. (4.7). Since hh_* has reciprocal roots, only those roots are included in h(z) that make it a hurwitz polynomial. For the z-domain, this implies that all the roots are strictly inside the unit circle. We now consider an example of embedding.

EXAMPLE 4.2

The following transfer function H(z) is to be embedded in a lossless matrix

$$H(z) = \frac{0.5(1 + 1.28\,z^{-1} + z^{-2})}{(1 + 0.8\,z^{-1} + 0.25\,z^{-2})}$$

Here, g and f are given as

$$g = 1 + 0.8\,z^{-1} + 0.25\,z^{-2} \qquad g_* = 0.25 + 0.8\,z^{-1} + z^{-2}$$

$$f = 0.5(1 + 1.28\,z^{-1} + z^{-2}) \qquad f_* = f \text{ (due to symmetry)}$$

Eqn. (4.7) gives

$$\begin{aligned} hh_* &= gg_* - ff_* \\ &= z^{-1}(.36 + .7929\,z^{-1} + .36\,z^{-2}) \\ &= [.75\,z^{-1}(.64 + z^{-1})][.75(1 + .64\,z^{-1})] \end{aligned}$$

Hence h is given by

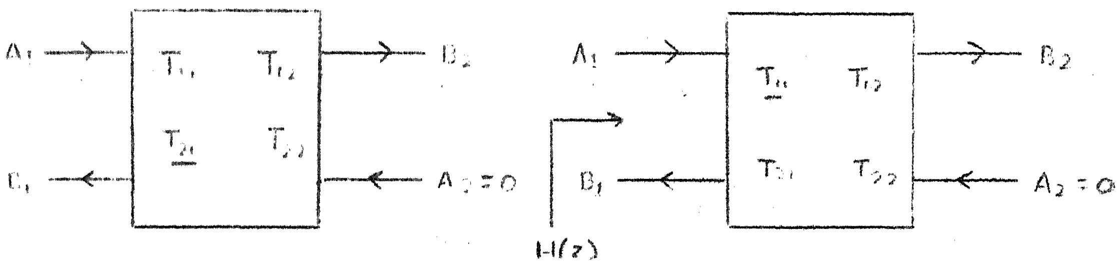
$$h = .75\,z^{-1}(0.64 + z^{-1})$$

The transfer matrix, therefore, is obtained as

$$T(z) = \frac{1}{(1+.8\ z^{-1}+.25\ z^{-2})} \begin{vmatrix} -.75(1+.64\ z^{-1}) & .5(1+1.28z^{-1}+z^{-2} \\ & \\ .5(1+1.28z^{-1}+z^{-2} & .75z^{-1}(0.64+z^{-1} \end{vmatrix}$$

It can be verified that $T^{t*}(z)\ T(z^{-1}) = I$. Thus $T(z)$ is lossless matrix.

As a variation, $H(z)$ can also be embedded as T_{11} as shown in Fig. (4.5b).



(a) T_{21} – Embedding (b) T_{11} – Embedding

Fig. 4.5 Embedding of transfer function.

After a lossy transfer function is embedded, into a lossless matrix, we should find means of implementing it as a digital 2-port whose overall description is as shown in Figs. (4.5). This is what we next discuss.

4.4 CASCADE IMPLEMENTATION OF LOSSLESS MATRIX

The lossless matrix obtained after embedding has the order which is same as the degree of the embedded transfer function. One way, therefore, is to realise this $T(z)$ by

4.5 LBR 2-PAIR EXTRACTION SYNTHESIS

This method aims at realising any structurally bounded or lossless bounded transfer function as a cascade of LBR 2-pairs. We start with a transfer function $H_m(z)$ and extract from it a LBR 2-pair in such a manner that the remaining transfer function $H_{m-1}(z)$ is of a lower degree. This is shown in Fig. (4.7).

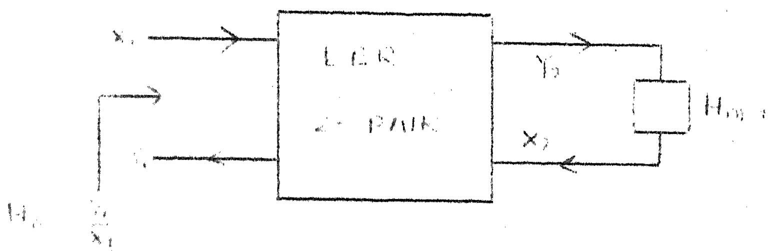


Fig. 4.7 Extraction of an LBR two-pair.

If the LBR 2-pair is described by its chain parameters, then the following relationship holds between $H_m(z)$ and $H_{m-1}(z)$.

$$H_m(z) = \frac{C+DH_{m-1}(z)}{A+BH_{m-1}(z)} \; ; \quad H_{m-1}(z) = \frac{C-AH_m(z)}{BH_m(z)-D} \tag{4.8a,b}$$

Next, we describe the form of first and second order LBR 2-pairs which are discussed in detail in [6].

A First Order LBR 2-Pair

The simplest form of LBR 2-pair which can be successively generated from given transfer function is of first order. The

chain and transfer parameters of this 2-pair are

$$A = 1 + \sigma z^{-1} \qquad T_{11} = \pm(1-\sigma)$$

$$B = \pm(1-\sigma)z^{-1} \qquad T_{12} = T_{21} = \sqrt{\sigma}\,(1+z^{-1})$$

$$C = \pm(1-\sigma) \qquad T_{22} = \mp(1-\sigma)z^{-1}$$

$$D = \sigma + z^{-1}$$

Common denominator $= \sqrt{\sigma}\;(1 + z^{-1})$ Common denominator $= 1 + \sigma z^{-1}$ (4.9a,b)

A point to note here is that the 2-pair described by parameter σ, has a transmission zero, i.e. $T_{21} = 0$, at $z^{-1} = -1$. Its importance will be shown later after we also describe a second order 2-pair which has transmission zeroes at $z^{-1} = e^{\pm j\omega_o}$.

Second Order LBR 2-Pair

When transmission occur in conjugate pairs say at $s = \pm j\omega_o$, then a 2-pair of second order is extracted. It has two parameters σ and β with $\beta = -\cos\omega_o$. The chain and transfer matrix has the following parameters

$$A = 1 + \beta(1+\sigma)z^{-1} + \sigma z^{-2} \qquad T_{11} = \pm(1-\sigma)(1+\beta z^{-1})$$

$$B = \pm(1-\sigma)z^{-1}(\beta+z^{-1}) \qquad T_{12} = T_{21} = \sqrt{\sigma}(1+2\beta z^{-1}+z^{-2})$$

$$C = \pm(1-\sigma)(1+\beta z^{-1}) \qquad T_{22} = \mp(1-\sigma)z^{-1}(\beta+z^{-1})$$

$$D = \sigma + \beta(1+\sigma)z^{-1} + z^{-2}$$

Common denominator

$$= \sqrt{\sigma}\,(1 + 2\beta z^{-1} + z^{-2})$$

Common denominator

$$= 1 + \beta(1+\sigma)z^{-1} + \sigma z^{-2}$$

(4.10a,b)

'One' Removal from the Transfer Function

To understand how an LBR 2-pair is extracted, we consider a first order pair. Its T_{11} and T_{12} parameters from eqn. (4.9b) are given as

$$T_{11} = \pm\frac{(1-\sigma)}{(1+\sigma z^{-1})}, \quad T_{12} = T_{21} = \frac{\sqrt{\sigma}(1+z^{-1})}{(1+\sigma z^{-1})}$$

Substituting $z^{-1} = -1$, we get $T_{11}(-1) = \pm 1$ and $T_{12}(-1) = T_{21}(-1) = 0$. Thus, at $z^{-1} = -1$, T_{11} has a 'one' while T_{12} has a zero. Output Y_1 is related to input X_1 and X_2 (figure 4.7) as

$$Y_1 = T_{11}(z)\,X_1 + T_{12}(z)\,X_2$$

Hence,

$$H_m(z)\Big|_{z^{-1}=-1} = \frac{Y_1}{X_1} = T_{11}(-1) + 0 = \pm 1$$

Eqns. (4.8b) and (4.9b) give $H_{m-1}(z)$ as

$$H_{m-1}(z) = \frac{\pm(1-\sigma) - (1+\sigma z^{-1})\,H_m(z)}{\pm(1-\sigma)z^{-1}\,H_m(z) - (\sigma+z^{-1})} \qquad (4.11)$$

If $H_{m-1}(z)$ is to have a degree one lower than $H_m(z)$ then we should have [ ]

$$\sigma = \left.\frac{H_m'(z^{-1})}{(H_m'(z^{-1}) \mp 1)}\right|_{z^{-1}=-1}, \quad 0 \leq \sigma \leq 1 \qquad (4.12)$$

Here superscript (') denotes derivative w.r.t. z^{-1}.

In short, degree reduction of the transfer function by means of extraction of a first order LBR 2-pair transmission zero at $z^{-1} = -1$ is equivalently expressed as a 'one' removal from the transfer function. Since, the transfer function is a structurally bounded function, sometimes a 'one' can be removed only at some frequency $\pm\omega_o$. In this case we extract a second order LBR 2-pair described by eqn. (4.10). Degree reduction is achieved if we let σ as

$$\sigma = \left.\frac{1}{1 \pm \frac{2z}{H_m'(z^{-1})}}\right|_{z = e^{j\omega_o}}, \quad 0 \leq \sigma \leq 1 \qquad (4.13)$$

In eqns. (4.9) and (4.10), lower signs are used if $H_m(-1) = -1$. Also, if $H_m(z)$ is unity at $z^{-1} = +1$ then all expressions can still be used after replacing z by z^{-1}.

We now consider some examples of the above synthesis procedure.

<u>EXAMPLE 4.3</u>

Consider the following lossless transfer function $H_2(z)$

$$H_2(z) = \frac{0.125 + 0.75\, z^{-1} + z^{-2}}{1 + 0.75\, z^{-1} + 0.125 z^{-2}}$$

Here we find that $H_2(-1) = +1$.

Therefore, a first order LBR 2-pair can be extracted. Taking the derivative, we get

$$H_2'(z) = \frac{0.65625 + 1.96875\ z^{-1} + 0.65625\ z^{-}}{(1 + 0.75\ z^{-1} + 0.125\ z^{-2})^2}$$

which gives,

$$H_2'(-1) = -14/3$$

Using eqn. (4.12) we obtain σ_2 as

$$\sigma_2 = \frac{-14/3}{-14/3 - 1} = 14/17$$

$H_1(z)$, by eqn. (4.11), is then given as

$$H_1(z) = \frac{-.0625 + z^{-1}}{1 - .0625^{-1}}$$

which is also a lossless function, as expected.

Once again, $H_1(-1) = -1$.

Repeating the procedure, we get $\sigma_1 = 15/31$

and $H_o = +1$.

The cascaded digital structure is shown in Fig. (4.8).

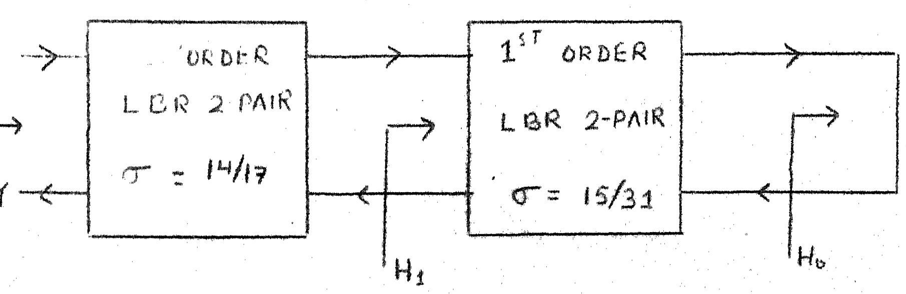


Fig. 4.8 Cascaded LBR two-pair structure of Example (4.3)

EXAMPLE 4.4

A structurally bounded transfer function $H_3(z)$ has value unity at $z^{-1} = -1$ and $e^{\pm j\pi/3}$. The function is given to be

$$H_3(z) = \frac{22 - 5z^{-1} + 5z^{-2} + 8z^{-3}}{24 - 5z^{-1} + 5z^{-2} + 10z^{-3}}$$

Suppose we first extract a 2nd order LBR 2-pair at $\omega_o = \pm \pi/3$. Eqn. (4.13) gives

$$\sigma = 1/4$$

Also $\beta = -\operatorname{Cos} \pi/3 = -1/2.$

$H_1(z)$ is obtained as,

$$H_1(z) = \frac{2 + z^{-1}}{3 + 2z^{-1}}$$

Note that extraction of a 2nd order 2-pair reduces the degree by two.

Substituting $z^{-1} = -1$, we get $H_1(-1) = +1$. Hence eqns. (4.11), (4.12) give

$$\sigma_1 = 1/2, \quad H_o = 1/3.$$

Fig. (4.9) shows the cascaded structure.

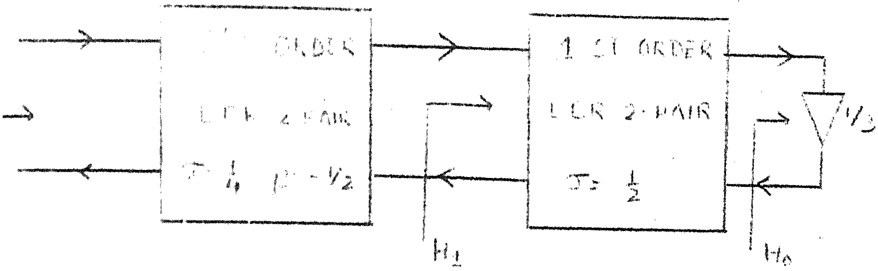


Fig. 4.9 Cascaded LBR two-pair structure of Example (4.4).

In the discussions on LBR 2-pair synthesis, we have assumed that the specific structure described by eqns. (4.9) and (4.10) is not sensitive to parameter σ. Before we discuss its implementation, we introduce yet another method of synthesizing low sensitivity digital filters. These are known as <u>cascaded orthogonal digital filters</u>. The name is so given because each building block is explicitly described by an orthogonal matrix.

4.6 CASCADED REALISATION OF ORTHOGONAL DIGITAL FILTERS

An alternative method to realise a low sensitivity digital filter is obtained by the direct factorization of the lossless transfer matrix. The lossy function which is embedded in this lossless matrix (section 4.3) is obtained by proper terminations at the ports as shown in Fig. (4.5). If the factorization is done into first or second order orthogonal matrices, then the structure obtained is that of the cascaded orthogonal filter introduced by Deprettere & Dewilde[5].

We present here only the basic ideas concerning these filters, leaving out details of design. These will be useful later while discussing the unified framework of all

low sensitivity digital filters.

Stability of Orthogonal Filters

Orthogonal filters, as outlined in [26], are digital filters whose internal computational scheme consists of an orthogonal matrix transformation. In other words, for the orthogonal filters, the realisation map from the input vector $\underline{X}$ to the output vector $\underline{Y}$ is represented by an orthogonal matrix.

It is well known that an orthogonal transformation preserves the norm of input vector. If [T] is an orthogonal transformation from $\underline{X}$ to $\underline{Y}$, i.e.

$$\underline{Y} = [T]\,\underline{X}$$

then output norm

$$\begin{aligned}\underline{Y}^{t*}\underline{Y} &= ([T]\,\underline{X})^{t*}([T]\,\underline{X}) \\ &= \underline{X}^{t*}[T]^{t*}[T]\,\underline{X} \\ &= \underline{X}^{t*}\underline{X} \quad (\text{Since } [T]^{t*}[T] = I).\end{aligned}$$

An advantage of [T] being an orthogonal matrix is that it can be represented by very few parameters. Consider, for example a 2x2 orthogonal matrix of the form

$$T(\phi) = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix} \qquad (4.14)$$

The matrix $T(\emptyset)$ in eqn. (4.14) involves only one parameter $\emptyset$. Moreover, it is observed that if this parameter is perturbed, say from $\emptyset$ to $\emptyset + \Delta\emptyset$, still the matrix $T(\emptyset + \Delta\emptyset)$ remains orthogonal. In other words, no instability is incurred atleast by the parameter quantization.

The above property is also used for a stable but very simple implementation of orthogonal transformations. The technique is known as the CORDIC processing technique and will be discussed later. But first we consider the low sensitivity aspects of orthogonal filters in the context of transfer function realisation.

Low Sensitivity of Orthogonal Filters

For lossless transfer matrix it can be shown that $T_{11}(z)$ and $T_{21}(z)$ have a complementary relationship, i.e.

$$\left|T_{11}(e^{j\omega})\right|^2 + \left|T_{21}(e^{j\omega})\right|^2 = 1 \qquad (4.15)$$

Differentiating eqn. (4.15), we get

$$\left|T_{11}(e^{j\omega})\right| \partial \left|T_{11}(e^{j\omega})\right| + \left|T_{21}(e^{j\omega})\right| \partial \left|T_{21}(e^{j\omega})\right| = 0 \qquad (4.16)$$

Suppose that the transfer function $H(z)$ is embedded as T_{21} in the lossless matrix. Then if $H(z)$ is a selective function which is close to zero in the stopband and close to unity in the passband (i.e. approximating Fig. (A.1)), then we have for the passband region,

$$\left|T_{21}(e^{j\omega})\right| \simeq 1$$

Because of eqn. (4.15) we also have

$$\left|T_{11}(e^{j\omega})\right| \simeq 0$$

Eqn. (4.16), therefore, gives

$$\left|T_{21}(e^{j\omega})\right| \simeq 0$$

This means that an arbitrary first order perturbation of any parameter ϕ not only preserves the orthogonality and hence stability but also leaves the behaviour of $H(e^{j\omega})$ unaffected, at least in the passband region.

Orthogonality plays an important role in the design and implementation of low sensitivity digital filters. In the next section, we will explain how an orthogonal matrix can be looked upon as a common building block of all low sensitivity structures. We also discuss the implementation of orthogonal matrices by planar rotations which are computationally stable.

SECTION FIVE

LOW SENSITIVITY STRUCTURES : A UNIFIED VIEWPOINT AND IMPLEMENTATION ASPECTS

5.1 INTRODUCTION

We have so far considered several methods of realising low sensitivity digital filters each of which seems to stand apart from others. These methods produce wave digital filters, Gray and Markel structures, cascaded LBR two-pairs and orthogonal digital filters. The realisation scheme of these methods differs so much, atleast apparently, from one another that it is natural to interpret each of them as giving a new low sensitivity structure. Yet all of these structures have a basic feature in common namely,that they consist of simple building blocks used as prototypes that are connected in a cascaded fashion. Moreover, a notion of losslessness is associated with each of these structures. In the case of WDFs and Gray and Markel's lattice filters, this losslessness is described in terms of pseudopower which is related to the incident and reflected signals. For LBR two-pairs, the transfer matrix in which the transfer function is embedded is lossless. Finally, for the orthogonal digital filters, each building block corresponds to an orthogonal transformation which preserves the norm of input vector.

In view of the above similarities, an attempt is made in [8] to find a link between various realisations mentioned earlier. In this section, we elaborate on this unified theoretical framework within which a close resemblance among these structures is clearly established. Later in the discussions, we consider certain issues connected with the implementation of low sensitivity digital filters. We discuss a stable but simple computational scheme known as CORDIC technique [27], that has been introduced for the orthogonal implementation. We also touch upon some fundamental concepts needed to understand VLSI implementation of these filters. A general synthesis procedure for VLSI design of these filters based on the work presented in [7] has also been included as an illustration.

5.2 SIMILARITY BETWEEN LOW SENSITIVITY STRUCTURES

The unified theory of all low sensitivity digital filters is based on the fact that the cascaded building blocks are always realisable by means of an orthogonal matrix. As we shall presently see, the transfer matrix in each case can be converted into an orthogonal matrix [8], without affecting the overall transfer function. We consider these structures one by one to show this orthogonal nature.

5.3 ORTHOGONAL IMPLEMENTATION OF LBR 2-PAIRS

As an illustration, we consider the first order LBR 2-pair with a transfer matrix T(z) having following parameters

$$T_{11} = \frac{(1-\sigma)}{(1+\sigma z^{-1})}, \quad T_{22} = \frac{-(1-\sigma)z^{-1}}{(1+\sigma z^{-1})} \tag{5.1}$$

$$T_{21} = T_{12} = \sqrt{\sigma}\,\frac{(1+z^{-1})}{(1+\sigma z^{-1})}$$

To realise this 2-pair as an orthogonal matrix, we first extract a delay element from it as shown in Fig. (5.1).

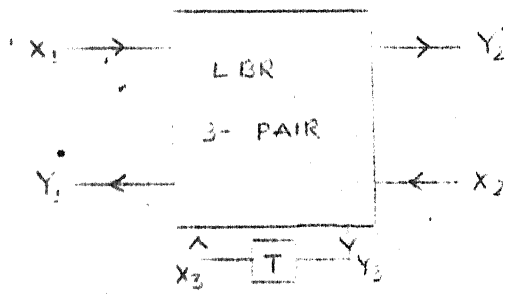


Fig. 5.1 Extraction of delay-element from an LBR 2-pair.

Figure (5.1) can be looked upon as a digital 3-pair whose port 3 is terminated in a delay element, i.e.

$$X_3 = z^{-1} Y_3 \tag{5.2}$$

Suppose that the digital 3-port has the following description

$$\begin{bmatrix} \underline{Y} \\ Y_3 \end{bmatrix} = \begin{bmatrix} [T_a] & \underline{T_b} \\ \underline{T_c}^t & T'_{33} \end{bmatrix} \begin{bmatrix} \underline{X} \\ X_3 \end{bmatrix} \tag{5.3}$$

where,

$$[T_a] = \begin{vmatrix} T'_{11} & T'_{12} \\ T'_{21} & T'_{22} \end{vmatrix}, \quad \underline{T_b} = [T'_{13}\ T'_{23}]^t, \quad \underline{T_c}^t = [T'_{31}\ T'_{32}]$$

also, $\underline{X} = [X_1\ X_2]^t$, $Y = [Y_1\ Y_2]^t$

Substituting eqn. (5.2) in (5.3), we obtain

$$Y_3 = \frac{1}{(1 - T'_{33}\, z^{-1})} \underline{T_c}^t \underline{X} \tag{5.4}$$

The transfer matrix T(z) is given as

$$T(z) = [T_a] + \frac{z^{-1}}{(1 - T'_{33}\, z^{-1})} \underline{T_b}\, \underline{T_c}^t \tag{5.5}$$

Equating term by term, eqns. (5.1) and (5.5), we get the following form of the transfer matrix T' of digital 3-pair,

$$T' = \begin{vmatrix} 1-\sigma & \sigma & \sigma \\ \sigma & 0 & -\sigma \\ -(1-\sigma) & \frac{-(1-\sigma)}{\sigma} & -\sigma \end{vmatrix} \tag{5.6}$$

It is observed from eqn. (5.5) that if $\underline{T_b}$ is scaled by a factor α and $\underline{T_c}$ a factor $1/\alpha$, then the transfer matrix T(z) is not affected.

If we choose $\alpha = \sqrt{\sigma/\ 1-\sigma}$ then eqn. (5.6) is rewritten as

$$T' = \begin{vmatrix} 1-\sigma & \sqrt{\sigma} & \sqrt{\sigma(1-\sigma)} \\ \sqrt{\sigma} & 0 & -\sqrt{1-\sigma} \\ -\sqrt{\sigma(1-\sigma)} & \sqrt{1-\sigma} & -\sqrt{\sigma} \end{vmatrix} \qquad (5.7)$$

One can verify that this matrix is orthogonal. Thus, a digital LBR 2-pair can be realised by an orthogonal matrix if the delay is extracted from it. Actual implementation of eqn. (5.7) will be discussed later. First we consider the case of wave digital filters.

5.4 ORTHOGONAL IMPLEMENTATION OF WAVE DIGITAL FILTERS

In wave digital filters, the adaptors are the building blocks of the ladder structure, as shown in Fig. (3.8). For example, a series adaptor (type S2) is given in Fig. (5.2). Port 3, which is terminated in a delay, corresponds to a

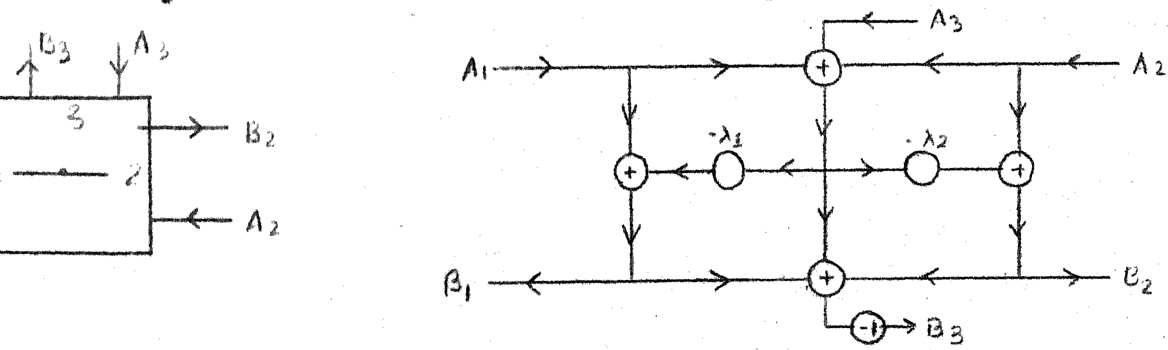


Fig. 5.2 A 3-port series adaptor.

shunt capacitor. Now if port 2 is reflection free, then the incident and reflected signals are related by

$$\begin{bmatrix} B_1 \\ B_2 \\ B_3 \end{bmatrix} = \begin{bmatrix} 1-\lambda_1 & -\lambda_1 & -\lambda_1 \\ -1 & 0 & -1 \\ -(1-\lambda_1) & -(1-\lambda_1) & \lambda_1 \end{bmatrix} \begin{bmatrix} A_1 \\ A_2 \\ A_3 \end{bmatrix} \qquad (5.8)$$

If port 2 wave quantities, A_2 and B_2, are scaled by a factor of $-\sqrt{\lambda_1}$ as shown in Fig. (5.3) and $-z^{-1}$ is extracted from port 2 instead of z^{-1} i.e.

$$A_3 = -z^{-1} B_3$$

and, $A_2 \rightarrow -A_2/\sqrt{\lambda_1}$, $B_2 \rightarrow -B_2\sqrt{\lambda_1}$

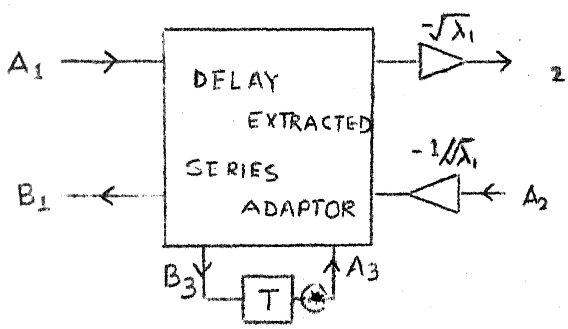


Fig. 5.3 Modified structure of series adaptor of Example (5.2).

then we have transfer matrix in eqn. (5.8) as,

$$\begin{bmatrix} 1-\lambda_1 & \sqrt{\lambda_1} & \lambda_1 \\ \sqrt{\lambda_1} & 0 & -\sqrt{\lambda_1} \\ -(1-\lambda_1) & \dfrac{(1-\lambda_1)}{\sqrt{\lambda_1}} & -\lambda_1 \end{bmatrix} \qquad (5.9)$$

Eqn. (5.9) is same as eqn. (5.6) with $\lambda_1 = \sigma$. Hence it can also be implemented as an orthogonal matrix as in eqn. (5.7) after proper scaling.

One can similarly show that a parallel adaptor too can be realised as an orthogonal matrix.

5.5 ORTHOGONAL IMPLEMENTATION OF CASCADED LATTICE FILTERS

We now consider the case of Gray and Markel filters which consist of a cascade of 2-port lattice or ladder structures. Chain matrix of this 2-port is given as

$$\begin{bmatrix} A_{m+1} \\ zB_{m+1} \end{bmatrix} = \begin{bmatrix} 1 & k_m \\ k_m & 1 \end{bmatrix} \begin{bmatrix} A_m \\ B_m \end{bmatrix} \tag{5.10}$$

Since the transfer function is realised by using ratio of B_m and A_m ($m = 1, \ldots, M$), chain parameters can be scaled by $\alpha = \sqrt{1-k_m^2}$ as shown in Fig. (5.4).

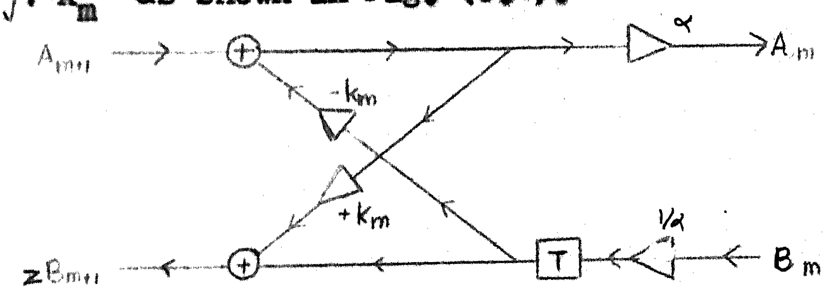


Fig. 5.4 Modified structure of ladder structure of Figure (4.3).

The scaled chain parameters are

$$A = D = 1/\sqrt{1-k_m^2}, \quad B = C = k_m/\sqrt{1-k_m^2} \tag{5.11}$$

Eqh. (5.11) gives the transfer matrix parameters as

$$T_{11} = k_m, \quad T_{22} = -k_m, \quad T_{12} = T_{21} = \sqrt{1-k_m^2}$$

Thus we can write,

$$\begin{vmatrix} B_{m+1} \\ A_m \end{vmatrix} = \begin{vmatrix} k_m & \sqrt{1-k_m^2} \\ \sqrt{1-k_m^2} & -k_m \end{vmatrix} \begin{vmatrix} B_m \\ A_{m+1} \end{vmatrix} \qquad (5.12)$$

Eqn. (5.12) is realised as in Fig. (5.5) and is known as the <u>normalised Gray and Markel ladder</u> because it is an orthogonal matrix.

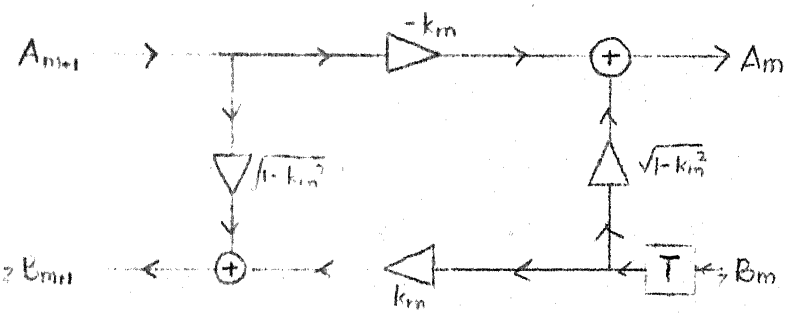


Fig. 5.5 Normalised Gray and Markel ladder structure.

Discussions in section (5.3) - (5.5) suggest that all of the considered low sensitivity digital filter structures can be realised as a cascade of blocks whose transfer matrices are orthogonal. It is the orthogonality of these matrices that makes us conclude the following. <u>Although different methods of designing low sensitivity digital filters give different structures, they essentially belong to one family of filters.</u>

In the next sub-section we look at some of the problem implementation of low sensitivity structures [23]. Conventionally, they can be implemented on any general purpose computer or a dedicated digital signal processor. These implementations provide flexibility in the design. Hardware implementation is however, preferred when speed is of prime consideration. Similarly, when mass production is required we need to design low sensitivity digital filters suitable for VLSI implementations. Before considering the VLSI design requirements, we shall see how an orthogonal matrix, the 'core' in all structure can be implemented by successive 'planar rotations'.

5.6 ORTHOGONAL MATRIX IMPLEMENTATION BY PLANAR ROTATIONS

As it was said earlier, an orthogonal transformation does not change the norm of input vector after transformation. It will be shown now that it merely 'rotates' the coordinate axes i.e. the output vector coordinates can be obtained by a simple rotation of the plane [27].

Consider for example eqn. (5.12). If we substitute $k_m = \cos \emptyset$, then the transfer matrix denoted by $T(\emptyset)$ is of the form

$$T(\emptyset) = \begin{bmatrix} \cos \emptyset & \sin \emptyset \\ \sin \emptyset & -\cos \emptyset \end{bmatrix} \qquad (5.13)$$

As shown in Fig. (5.6), this matrix can be implemented as a rotation of the plane formed by axes X_1 and X_2.

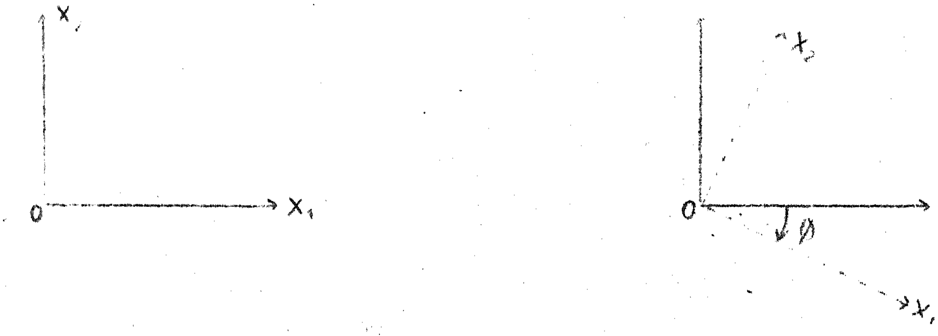


Fig. 5.6 Implementation of orthogonal matrix by planar rotation.

Similarly by substituting σ (λ_1 for WDFs) = cos ø in eqn. (5.7), we obtain the transfer matrix T(ø) as

$$\begin{vmatrix} \sin^2 \phi & \cos \phi & \sin \phi \cos \phi \\ \cos \phi & 0 & -\sin \phi \\ -\sin \phi \cos \phi & \sin \phi & -\cos^2 \phi \end{vmatrix} \tag{5.14}$$

This orthogonal matrix can be factorized into a product of two simpler orthogonal matrices, each of which corresponds to a planar rotation similar to the one shown in Fig. (5.6). The factorization is as follows

$$T(\phi) = T_2(\phi)\, T_1(\phi) = \begin{Vmatrix} \cos \phi & \sin \phi & 0 \\ 0 & 0 & 1 \\ \sin \phi & -\cos\phi & 0 \end{Vmatrix} \begin{Vmatrix} 0 & 1 & 0 \\ \sin \phi & 0 & \cos \phi \\ \cos \phi & 0 & -\sin \phi \end{Vmatrix} \tag{5.15}$$

Fig. (5.7) shows how eqn. (5.15) can be looked upon as a combination of two planar rotations.

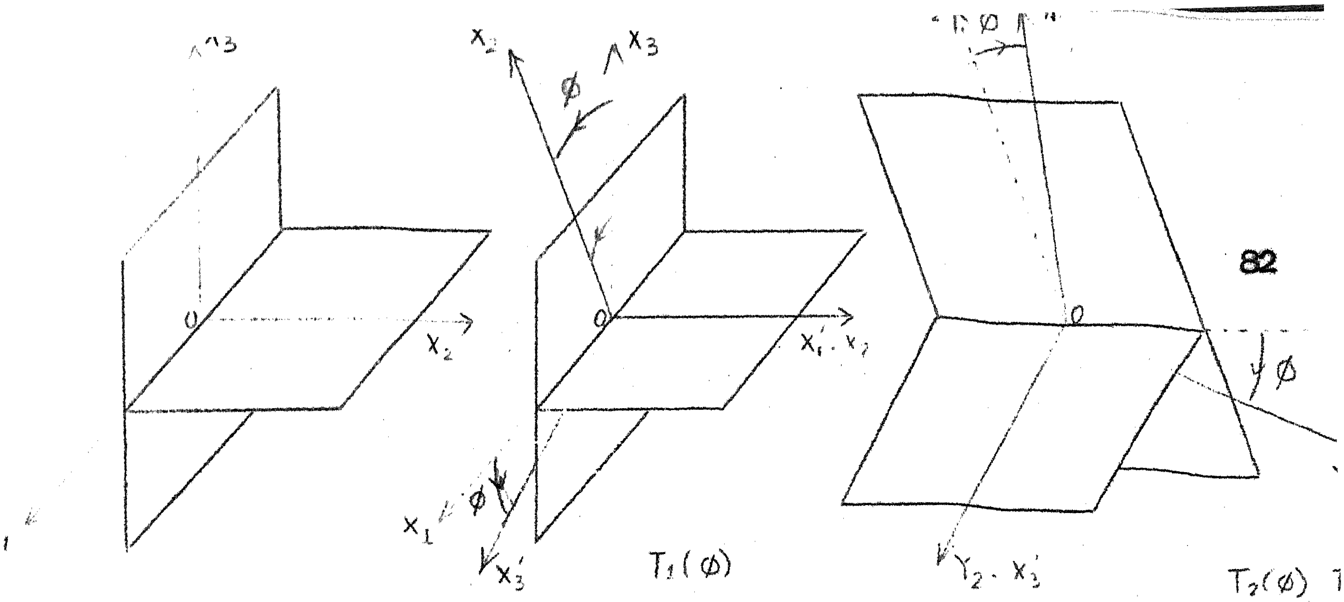


Fig. 5.7 Successive planar rotations for a general orthogonal matrix.

In section (4.6) it was shown that the orthogonality of an orthogonal matrix is not disturbed by any first order perturbations of the parameter ϕ. This results in the stable configuration of orthogonal implementation of filters. We now consider an algorithm known as CORDIC algorithm which is very suitable for implementation of orthogonal matrix by successive rotations.

5.7 THE 'CORDIC' ALGORITHM

The primary purpose of this algorithm [27] is to compute the output vector $\underline{B}$ which is obtained by an orthogonal transformation, $T(\alpha)$, of the input vector $\underline{A}$. This is given by

$$\begin{bmatrix} X_n \\ Y_n \end{bmatrix} = \begin{bmatrix} T(\alpha) \end{bmatrix} \begin{bmatrix} X_o \\ Y_o \end{bmatrix}$$

where,

$$\underline{B}^t = [X_n \; Y_n]^t, \; \underline{A}^t = [X_o \; Y_o]^t \quad \text{and} \quad T(\alpha) = \begin{vmatrix} \cos\alpha & -\sin\alpha \end{vmatrix}$$

The computation for this process is carried out by successive planar rotations of the input vector. The salient feature of the algorithm is that only shifts and 'cross-additions' are required. This results in the numerical stability of the computational scheme.

The algorithm is as follows. Suppose, that α is the angle by which input vector $\underline{A}$ is to be rotated to get the output vector $\underline{B}$. Let α be expressed as a linear combination of angles $\alpha_1, \alpha_2 \ldots \alpha_n$:

$$\alpha = \sum_{i=0}^{n} \lambda_i \alpha_i \tag{5.16}$$

Here λ_i can be $\pm 1, 0$ for $i = 1$ to n and they are also called the Arc tangent radices (ATR) digits.

If n is the length of the memory register and each bit denotes an angle α_i, such that

$$\alpha_i = \mathrm{Tan}^{-1}(2^{-(i-1)}), \quad i > 0 \tag{5.17}$$

then each nonzero bit denotes an angle α_i.

Also, $\alpha_o = 90^o$.

Consider now the situation shown in Fig. (5.8) where a vector OA_i is rotated by an angle $\pm \alpha_i$.

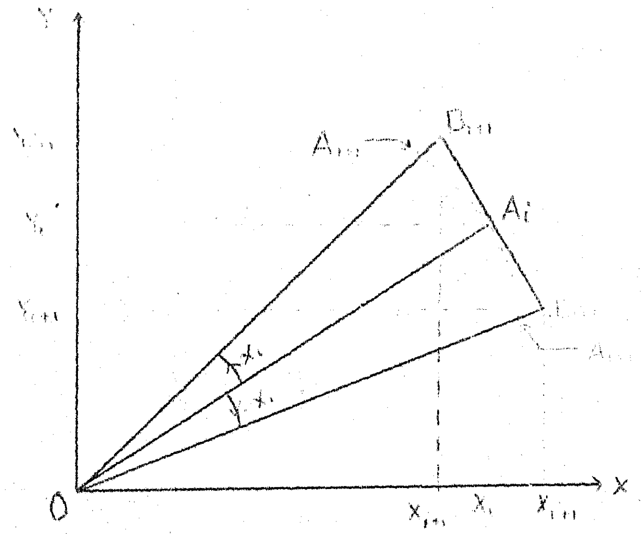


Fig. 5.8 CORDIC approximation of planar rotation.

If angle α_i is small enough then the dotted arc can be approximated by a straight line. In such a case, coordinates of B_{i+1} are given as

$$X_{i+1} = X_i \mp 2^{-(i-1)} Y_i$$

$$Y_{i+1} = Y_i \pm 2^{-(i-1)} X_i \qquad (5.18)$$

Eqn. (5.18) can be very easily implemented by using simple <u>shift</u> and <u>cross add</u> operations. The typical configuration for this purpose is shown in Fig. (5.9).

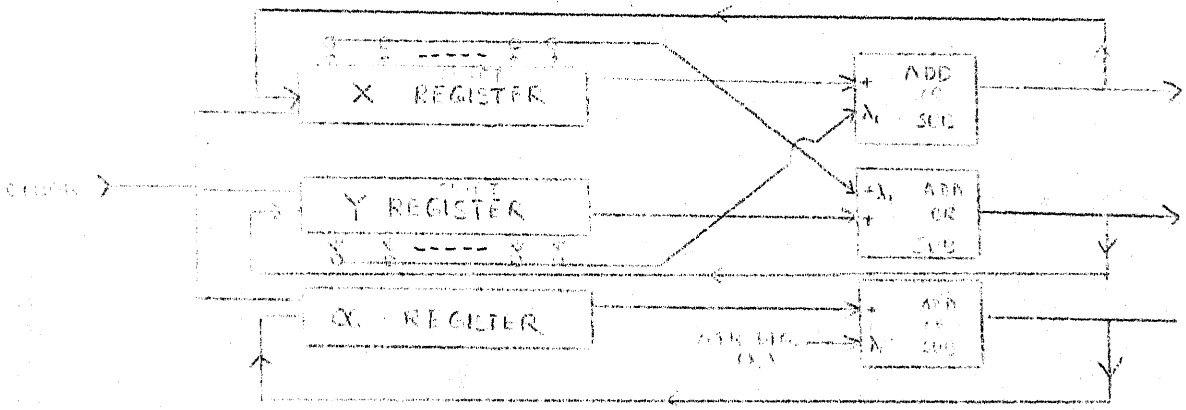


Fig. 5.9 General structure of CORDIC implementation.

It is noted in Fig. (5.8) that any rotation α_i, the magnitude of vector OA_i is increased by the factor $1+2^{-(i-1)2}$.

Since length n of the re ister is fixed the overall incremental error denoted by k is a constant. This is given as

$$k = \prod_{i=1}^{n} \sqrt{(1 + 2^{-(i-1)2})} \qquad (5.19)$$

To illustrate the algorithm we consider the following example.

EXAMPLE 5.1

Compute the coordinates of P' obtained after rotating P by 19° counterclockwise using CORDIC algorithm for 8 bit implementatio

Here, using eqn. (5.17) α_i's are given as

$\alpha_0 = 90^\circ$, $\alpha_1 = 45^\circ$, $\alpha_3 = 14.036^\circ$

$\alpha_4 = 7.125^\circ$, $\alpha_5 = 3.5763^\circ$, $\alpha_6 = 1.290^\circ$

$\alpha_7 = 0.895^\circ$, $\alpha_8 = 0.448^\circ$.

For $\alpha = 19^\circ$ we can have

$\lambda_2 = +1$, $\lambda_4 = -1$, $\lambda_8 = -1$

$\lambda_0 = \lambda_1 = \lambda_3 = \lambda_5 = \lambda_6 = \lambda_7 = 0$, which gives

$\alpha = 18.9925$ the computation is carried out as follows :

i	λ_i	X_i	Y_i
0	0	.01100000	.01100000
		−.00110000	+.00110000
2	+1	.00110000	.10010000
		+.00010010	−.00000110
4	−1	.01000010	.10001010
		+.00000001	−.00000000
8	−1	.01000011	.10001010

Thus the computed result is (.2617, .5390) where as the actual result is (.2394, .4767).

We have just seen that it is possible to implement an orthogonal matrix by a very stable planar rotation method. This method is also suitable for VLSI implementation on account of its 'modularity'. In the following discussions we look at some of the requirements that need to be met for the VLSI implementation of these filters.

5.8 REQUIREMENTS FOR VLSI DESIGN

With the recent advances in microelectronics, certain digital filter structures that can be realised on a single chip

have become more attractive than others. Certain terms need to be defined before one can appreciate the new requirements for VLSI implementation of these filters.

Modularity (Regularity)

The most important consideration for a structure to be VLSI implementable is that of modularity. In other words, the entire structure must be a regular interconnection of similar processing elements. It is also called an array type architecture and can be easily fabricated on a single chip.

Local Connectivity (Systolic Nature)

One prefers in VLSI design those structures which can be drawn in a single plane with no 'crosswires'. If in the entire array, each processor has only the nearest neighbour links, the structure is called systolic in nature as shown in Fig. (5.10).

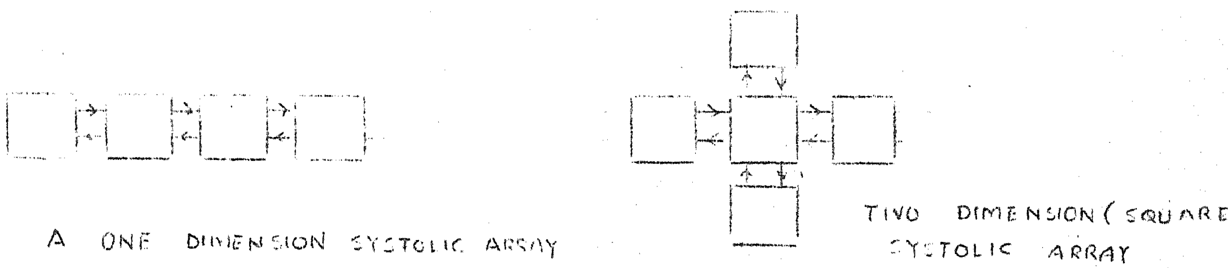


Fig.5.10 Examples of systolic structures.

Pipelineability

array type structure, the processing time for each input data depends on the length of array even though it is systolic in nature. This is because of the fact that the input to any processing element is available only after processing by its adjacent element(s) is completed. Such structures are called non-pipelineable.

A pipelineable digital filter is one in which the maximum throughput rate (A.12) does not depend on the order of filter which is directly proportional to length of the array.

Uniformity

It is also desirable to design VLSI structures that can implement digital filters with a variety of transfer functions of different orders merely by changing certain parameters. Such structures are called uniform in nature and can be used to process several input sequences for different transfer functions. This results in increased throughput of the structure and increased 'active' time of each processing element.

Having introduced some important concepts of VLSI implementation, we consider the suitability of available low sensitivity structures for the VLSI design viewpoint.[28].

5.9 VLSI CONSIDERATION FOR LOW SENSITIVITY DIGITAL FILTERS

All structures of low sensitivity digital filters

This includes wave digital filters also even though they are realised by an indirect method. Systolic nature too is present in each of these filter structure.

Pipelineability, however, needs to be investigated further. For WDFs, in which parallel or series adaptors are alternately connected (Fig. 3.8), outputs of any adaptor is computed from its inputs. These inputs inturn depend on current outputs of adjacent adaptors. As a result, the structure is non-pipelineable. Hence, the larger is the order of filter transfer function, more is the processing time for an input data. On the other hand, Gray and Markel and LBR 2-pair structures are pipelineable. This is so because of a delay introduced between successive blocks. Thus processing time for an input sample is the same as that of one processing element. Such structures are also called <u>synchronous</u> because each element works in synchronism with others.

None of the available two sensitivity structures is, however, uniform in nature. This is so because given a transfer function, by merely reodering the cascaded blocks one can not achieve different realisations of the design with different parameters. In the next sub-section, we present an orthogonal digital filter structure which is both pipelineable and uniform.

5.10 PIPELINEABLE AND UNIFORM ORTHOGONAL FILTERS

In a recent work [729] a scheme, to be presented here, has been discussed which has all the features of VLSI

requirements. In this design the basic building block has a 3x3 transfer matrix and is shown in Fig. (5.11).

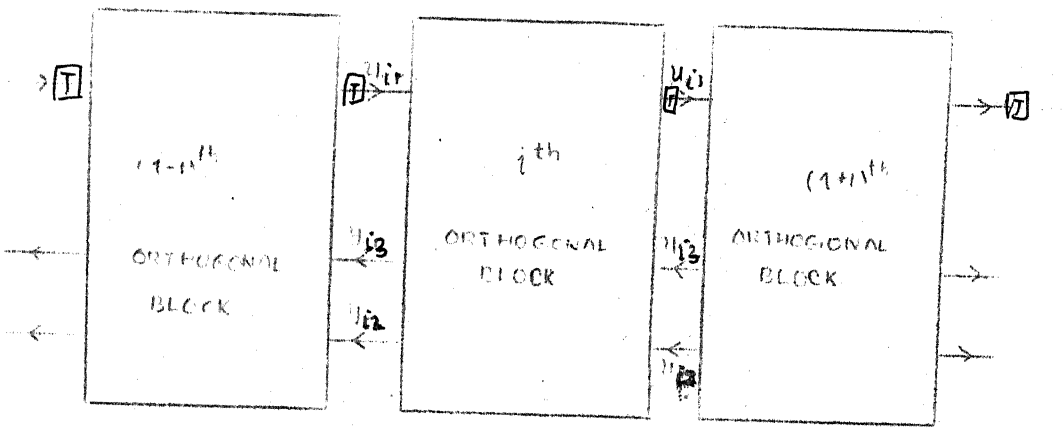


Fig. 5.11 Structure of an orthogonal digital filter.

For such filter the transfer matrix T(z) is orthogonal and the input output relationship is given by

$$\begin{bmatrix} y_{i2}(z) \\ y_{i3}(z) \end{bmatrix} = \frac{1}{Q(z)} \begin{bmatrix} P(z) \\ R(z) \end{bmatrix} u_{i1}(z) = T(z).\, u_{i1}(z) \tag{5.20}$$

the orthogonality of T(z) gives

$$PP_* + RR_* = QQ_* \tag{5.21}$$

where $A_*(z) = A(1/z)$.

Now, for a given transfer function H(z), there are several choices of T(z). Suppose we wish to use

$$u(z) = k_1 u_{i1}(z) + k_2 y_{i2}(z)$$
$$y(z) = y_{i3}(z) \tag{5.22}$$

Such that H(z) is the transfer function from u(z) to y(z). In this case, the following relationship holds

$$P(z) = N(z)$$
$$k_1 Q(z) + k_2 R(z) = D(z) \tag{5.23}$$

Here N(z), D(z) are the numerator and denominator polynomials respectively of transfer function H(z). Equation (5.23) suggests that by merely changing parameters k_1 and k_2, different transfer functions can be obtained. Further, the same transfer function can also be realised in more than one way. This results in the 'uniform' nature of the filter structure as described earlier. For illustration, we consider an example.

EXAMPLE 5.2

Realise a transfer function H(z) = p(z)/e(z) for the followi cases

(i) $k_1 = 1$, $k_2 = 0$ and (ii) $k_1 = k_2 = 1$

Case I

Substituting $k_1 = 1$ and $k_2 = 0$ in eqn. (5.23), we get

$P(z) = p(z)$

$Q(z) = e(z)$

If we replace R in eqn. (5.21) by f then we can write

$ee_* = pp_* + ff_*$

which is same as the feldkeller eqn. (1.2) for which standard solutions are available if H(z) is a generalised transfer function (A.1).

(ii) Case II

Substituting $k_1 = k_2 = 1$ in eqn. (5.23), one obtains

$P(z) = p(z)$

$$Q(z) + R(z) = e(z) \tag{5.24}$$

If we choose $Q(z) = (e(z) + f(z))/2$, $R(z) = (e(z) - f(z))/2$ then substituting in eqn. (5.24) we get

$ef_* + fe_* = 2pp_*$

Implementation of Transfer Matrix

Once T(z) is obtained using eqns. (5.20 - 5.23), it is possible to factorize it by an algorithm (A.13) into simple orthogonal building blocks [7] as shown in Fig. (5.11). The chain matrix Θ_i of each block can be written as a product of

$\theta_i = \theta_{i2} * \theta_{i1}$, where θ_{ij} $(j = 1,2)$ are given by

$$\theta_{i1} = \frac{1}{\sqrt{1-k_{i1}^2}} \begin{bmatrix} 1 & -k_{i1} & 0 \\ -k_{i1} & 1 & 0 \\ 0 & 0 & \sqrt{1-k_{i1}^2} \end{bmatrix},$$

$$\theta_{i2} = \frac{1}{\sqrt{1-k_{i2}^2}} \begin{bmatrix} 1 & 0 & -k_{i2} \\ 0 & \sqrt{1-k_{i2}^2} & 0 \\ -k_{i2} & 0 & 1 \end{bmatrix} \qquad (5.25)$$

Where k_{ij}'s $(j = 1,2)$ are obtained as in algorithm (A.13). Fig. (5.12) shows how each θ_i can be realised as a cascade of two normalised Gray and Markel's ladder structures of Fig. (5.5). The pipelineability of the structures follows from the fact that a delay element is present between any two building blocks (Figure (5.11)). Equations (5.25) can be implemented by two planar rotations. As we shall presently see, a different form of CORDIC algorithm implementation results in the processing of parallel sequences in this structure.

Parallel Processing of Input Sequences

As stated earlier in the CORDIC algorithm on angular relation is achieved through successive steps, say n in number.

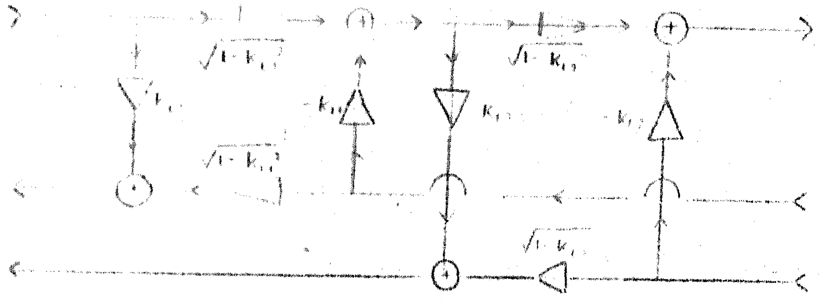


Fig. 5.12 Building block of the orthogonal digital filter.

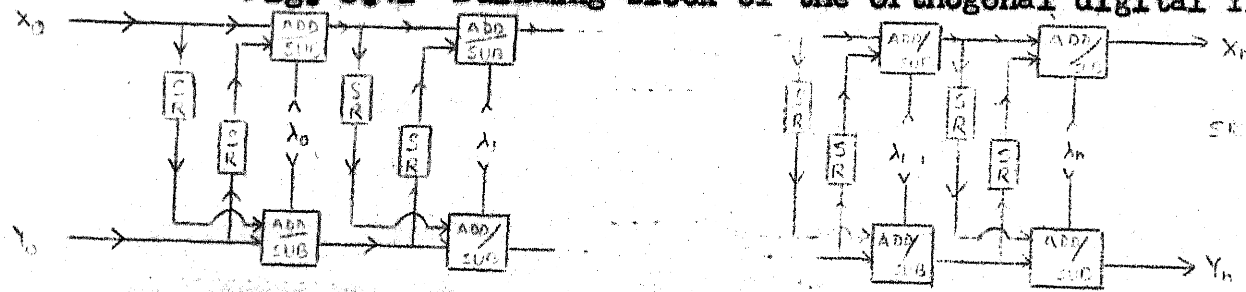


Fig. 5.13 Forward going implementation of CORDIC processor.

Instead of the configuration of Fig. (5. 9), if we use 'forward going' shifts and cross-adders as shown in Fig. (5.13), we infer that each 'shift-adder' is active only for a fraction (1/n) of the entire processing period T. Hence if buffer registers of length n are used for parallel input sequences and ATR constants for different transfer functions are stored in a circular registers of length n, then it is possible to process several input sequences in parallel.

In the discussions, we have shown how different structures of low sensitivity digital filters are realisable using orthogonal building blocks. These blocks are implemented in a stable manner by successive planar rotations such that it is least sensitive to the perturbations in the parameter values. In a broader perspective, theory of low sensitivity digital filters is also able to offer competent designs for the VLSI implementation of digital filters which is not guaranteed by the conventional design methods.

SECTION SIX

CONCLUDING REMARKS

In this work we have discussed the design of low sensitivity digital filters using different kinds of structures. Simulation results (Appendix B) for these structures have been shown to demonstrate their superior performance over conventional methods of implementing IIR digital filters. It has also been shown that a general theory based on the concept of orthogonality provides a more concrete and satisfactory explanation for the working of the various low sensitivity structures. We have observed that many of these structures are also suitable for VLSI implementation. Based on these facts, it is reasonable to conclude that low sensitivity implementations offer a more attractive alternative to conventional schemes in most hardware applications.

It was originally envisaged that the design of FIR digital filters will also be investigated. For these filters, the realisation map from input sequence to output sequence can be considered as a linear transformation characterized by a cyclic transfer matrix. It would, therefore, be interesting to see if it is possible to impose certain conditions on the matrix, say orthogonality, for a low sensitivity implementation of 'block' input sequences. One may possibly be able to obtain vector realisation structures analogous to the parallel

processing structures for the IIR filters. This line of investigation, however, could not be pursued further owing to lack of time and a ready access to some of the recent related literature.

APPENDIX A

A.1 GENERALISED FILTER FUNCTIONS

Generalized filter functions are used to approximate the ideal frequency response. An ideal low pass frequency response is shown in Fig. (A.1).

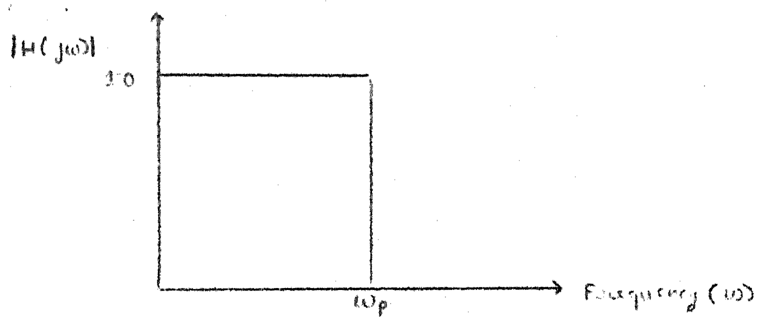


Fig. A.1 An ideal lowpass frequency response.

The transmittance of a filter is expressed in terms of the generalized filter function $K(j\omega)$ as

$$|t(j\omega)|^2 = \frac{1}{1 + |K(j\omega)|^2}$$

Butterworth Filters

For n^{th} order Butterworth filter,

$$|K(j\omega)|^2 = C^2\omega^{2n} \qquad (A.1)$$

where n : order of the filter, C : a constant

or,

$$f(s) = \pm Cs^n, \quad p(s) = 1 \qquad (A.2)$$

If A_p and A_s are maximum and minimum passband and stopband attenuations respectively then,

$$n \geq \frac{\mathrm{Log}(1/K_1)}{\mathrm{Log}(1/K)} \tag{A.3}$$

where

$$K_1 = \sqrt{\frac{10^{.1A_p}-1}{10^{.1A_s}-1}} \quad ; \quad K = \omega_p/\omega_s \tag{A.4a,b}$$

Using the least value of n and meeting the stopband requirement strictly,

$$C = \sqrt{(10^{.1A_p}-1)}/\omega_p^n \tag{A.4c}$$

Solving for e(s) gives, (considering only LHP zeroes for stability)

$$e(s) = \pm C \prod_{k=1}^{n} (s-s_k) \quad ; \quad s_k = C^{-1/n} \exp\left(j\left(\frac{\pi}{2} + \pi \frac{2k-1}{2n}\right)\right)$$

$$k = 1,\ldots,n \tag{A.5}$$

Chebyshev Filters

For n^{th} order Chebyshev filter,

$$|K(j\omega)|^2 = C^2 T_n^2(\omega) \tag{A.6}$$

where $T_n(\omega)$ is the n^{th} order Chebyshev function

$$[T_o = 1,\ T_1 = \omega \ ; \ T_n(\omega) = T_{n-1}(\omega).2\omega - T_{n-2}(\omega)]$$

$$n \geq \frac{\mathrm{Cosh}^{-1}(1/K_1)}{\mathrm{Cosh}^{-1}(1/K)} \tag{A.7}$$

where K_1, K and C are given by (A.4a,b,c) above

$$e(s) = \pm C_o \prod_{k=1}^{n} (s-s_k),\ s_k = \sigma_k + j\omega_k \tag{A.8}$$

$$\sigma_k = \mathrm{Sinh}\left(\frac{1}{n}\,\mathrm{Sinh}^{-1}\frac{1}{C}\right) \mathrm{Sin}\left(\frac{2k-1}{2n}\,\pi\right)$$

$$\omega_k = \text{Cosh}\left(\frac{1}{n}\,\text{Sinh}^{-1}\,\frac{1}{C}\right)\,\text{Cos}\left(\frac{2k-1}{2n}\,\pi\right)$$

for k = 1 to n. C_o is an appropriate constant.

A.2 POSITIVE REAL FUNCTIONS (PRF)

A polynomial with its roots strictly in the LHP is a <u>hurwitz</u> polynomial. If in addition to being hurwitz it is permitted to have simple zeroes on the $j\omega$ axis then it is <u>modified hurwitz</u>. A positive real function is one which has following properties

(a) It has rational coefficients

(b) Denominator is hurwitz or modifed hurwitz

(c) Real part of function (with $s = j\omega$) is non-negative

(d) Residues of the $j\omega$-axis poles are real and positive.

EXAMPLE A.1

Check if $Z(s) = \dfrac{3s+4}{s^2+2s}$ is a PRF.

(a) is obvious. $s = -2$ is in LHP and $s = 0$ is a simple root on the ω axis. Thus denominator is modified hurwitz and (b) is satisfied. Also,

$$R_e(Z(j\omega)) = \frac{\omega^2}{\omega^4+4\omega^2}\,;$$ which is non negative, being even function with positive coefficients. Hence, (c) is satisfied.

Residue at $s = 0 = \lim_{s\to 0} sZ(s) = 2 > 0$, satisfying (d)

hence Z(s) is a PRF.

In network theory, PRFs derive their importance from the result that a function is positive real iff it is realizable as an immittance of an RLCM network.

A.3 REALISATION OF NETWORK FUNCTIONS

Foster's Synthesis

In this method, an immittance function is realised by successive pole removals. First, all poles on $j\omega$ axis including those at $s = 0$ and $s = \infty$ are removed. This is given by equation (A.9) and gives Foster's synthesis if $F_I(s)$ is zero.

$$F(s) = \frac{K_o}{s} + K_\infty s + \sum_{i=1}^{m} \frac{2K_i s}{s^2+\omega_i^2} + F_I(s) \qquad (A.9)$$

where K_i are residues. The network is realised as shown in Fig. (A.2).

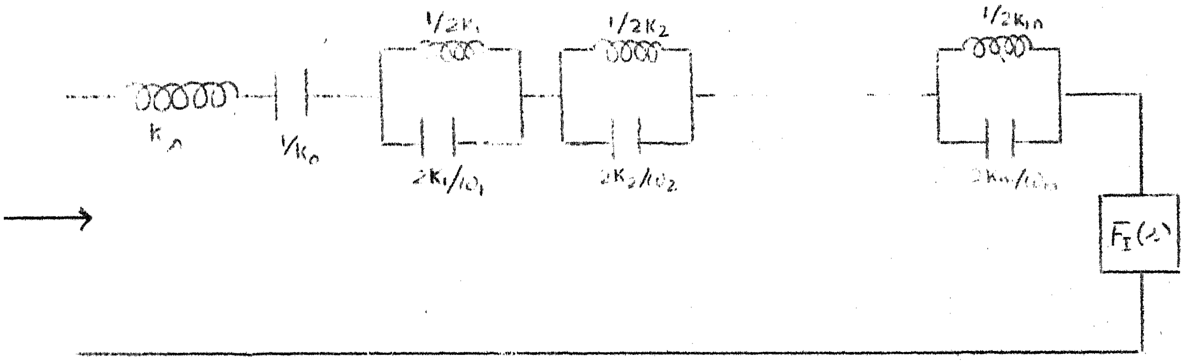


Fig. A.2 A Foster's network.

$F_I(s)$ is called the minimum function since it can not be reduced further by extracting either an inductance or a capacitance

It is realised by removing a constant so that the remaining part is still a PRF. This is illustrated by an example.

EXAMPLE A.2

$$Z(s) = \frac{18s^2 + 7s + 2}{6s + 1}$$

It has a pole at $s = \infty$

$$K_\infty = \lim_{s \to \infty} \frac{18s^2 + 7s + 2}{s(6s + 1)} = 3$$

After removal of this pole we have,

$$Z_I(s) = Z(s) - 3s = \frac{4s + 2}{6s + 1}$$

Since $Z_I(o) = 2$ and $Z_I(\infty) = 2/3$, we remove a minimum constant $Z_I(\infty)$. This leaves

$$Z_I(s) = \frac{4s + 2}{6s + 1} - \frac{2}{3} = \frac{1}{\frac{9}{2}s + \frac{3}{4}}$$

Hence Z(s) is realised as in Fig. (A.3).

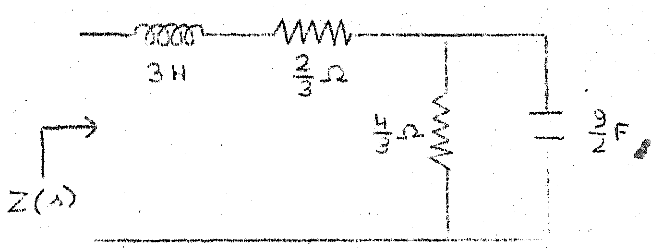


Fig. A.3 Foster's network of Example (A.2).

Cauer's Synthesis

In the Cauer's realisation, the PRF is reduced by one

degree, each time a pole at $s = \infty$ is removed either from $Z(s)$ or from $Y(s)$. It generates a ladder as shown in Fig. (A.4).

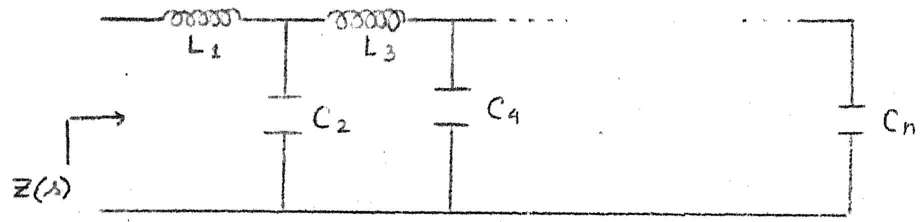


Fig. A.4 A Cauer's ladder network.

EXAMPLE A.3

Consider $$Z(s) = \frac{s(s^2+4)(s^2+16)}{(s^2+1)\,(s^2+9)}$$

Removing a pole at $s = \infty$

$$Z(s) = s + \frac{10s^3 + 55s}{s^4 + 10s^2 + 9} = s + Z_1(s)$$

Removing a pole at $s = \infty$ from $Y_1(s)$

$$Y_2(s) = Y_1(s) - \frac{1}{10}s = \frac{9s^2 + 18}{20s^3 + 110s}$$

Repeating this procedure we get the entire network as shown in Fig. (A.5).

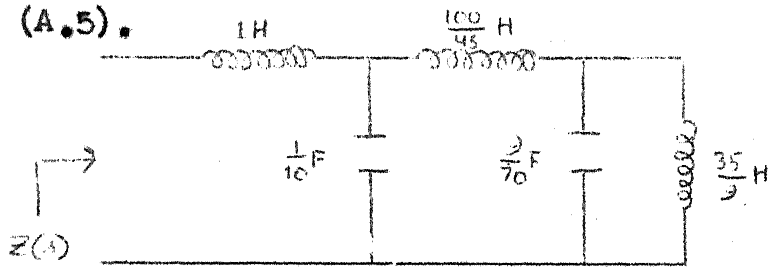


Fig. A.5 Ladder network of Example (A.3).

A.4 LATTICE NETWORK RELATIONSHIPS

For a symmetric lattice network

$$Z_{11} = Z_{22} = \frac{1}{2}(Z_a + Z_b)$$

$$Z_{12} = Z_{21} = \frac{1}{2}(Z_b - Z_a)$$

Using terminating conditions for $R_1 = R_s = R$

$$V_1 + I_1R = V_s$$

$$V_2 + I_2R = 0$$

We obtain

$$V_2 = \frac{(Z_b - Z_a)}{(2R + Z_a + Z_b)} I_1 \qquad (A.10)$$

and

$$V_s = \left(\frac{(2R+Z_a+Z_b)^2 - (Z_a - Z_b)^2 R}{2(2R+Z_a+Z_b)}\right) I_1 \qquad (A.11)$$

Substituting eqns. (A.10) and (A.11) in the expression for $t(s)$ we get

$$U(s) = 1/t(s) = \frac{1}{2}\frac{V_s}{V_2} = \frac{(R+Z_a)(R+Z_b)}{R(Z_b - Z_a)}$$

Since Z_a and Z_b are lossless and hence odd functions, we have

$$U(s)\cdot U(-s) - 1 = \left(\frac{(R+Z_a)(R+Z_b)}{R(Z_b - Z_a)}\right)\left(\frac{(R-Z_a)(R-Z_b)}{R(Z_a - Z_b)}\right) - 1$$

$$= \frac{-(R^2-Z_bZ_a)^2}{R^2(Z_b-Z_a)^2}$$

$$= K(s)\cdot K(-s)$$

where $$K(s) = \frac{(R^2 - Z_bZ_a)}{R(Z_b - Z_a)}$$

From above equation we get

$$U_e = \frac{Z_a+Z_b}{Z_b-Z_a}, \quad U_o = \frac{Z_aZ_b+R^2}{Z_b-Z_a}, \quad K_o = \frac{R^2 - Z_bZ_a}{R(Z_b - Z_a)}$$

which give the expressions for Z_a and Z_b in terms of U_e, U_o and K_o.

A.5 STABILITY OF MICROWAVE FILTERS

For $s = j\omega$,

$$\psi = \mathrm{Tanh}(j\omega T/2)$$

$$= j\mathrm{Tan}(\omega T/2) = j\phi$$

A plot of ω versus ϕ is shown in Fig. (A.6).

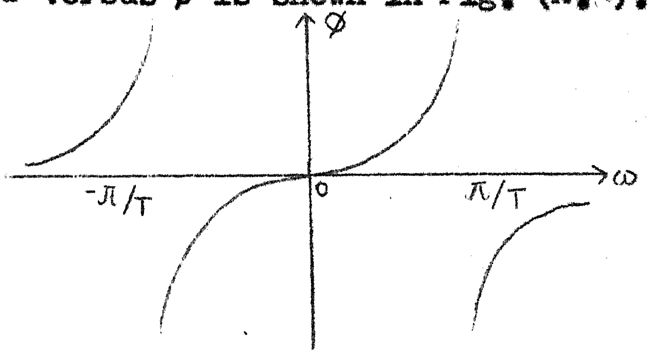


Fig. A.6 A plot of ω vs ϕ.

Thus the entire ψ-plane is mapped onto a strip from $-\pi/T$ to $+\pi/T$ in the s-plane. Moreover,

$$0 \leq \omega \leq \pi/T \quad \Longrightarrow \quad 0 \leq \emptyset \leq \infty$$

and, $$-\pi/T < \omega < 0 \quad \Longrightarrow \quad -\infty < \emptyset < 0$$

therefore, a pole in LHP of plane is mapped in the LHP strip in the s-plane preserving the stability condition in the s-plane.

A.6 RICHARD'S THEOREM

If Z(s) is a PRF and Z(s)/Z(1) is not equal to s or 1/s, then a unit element of impedance Z(1) can always be extracted from Z(s), leaving a PRF $Z_1(s)$. If $Z(1) = Z(-1)$, then $Z_1(s)$ is lower by one degree. $Z_1(s)$ is given by

$$Z_1(s) = Z(1)\,\frac{Z(s)-sZ(1)}{sZ(s)-Z(1)} \qquad (A.12)$$

The theorem holds for admittance function Y(s) as well with Z replaced by Y.

A.7 DELAY FREE LOOPS

Delay free directed loops can be generated while interconnecting an adaptor port to an element or a port of another adaptor. Both capacitance and inductance involve a delay element. Similarly a resistance is an open circuit. Thus delay free loops are never generated whenever any port is connected to RLC elements. However, they may be present when a port is terminated by a 'short' or 'open' circuit or connected to another adaptor as shown in Fig. (A.7).

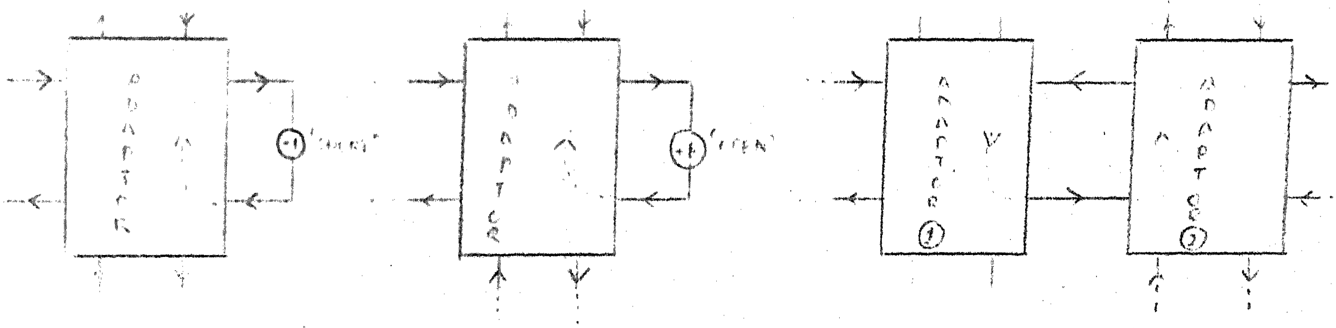


Fig. A.7 Generation of delay free lopps.

To avoid these delay free loops, we can make one of the ports to be connected as reflection free. Thus for an n-port adaptor if we choose

$$R_n = R_1 + R_2 + \dots R_{n-1} \quad (A.13a)$$

or,

$$G_n = G_1 + G_2 + \dots G_{n-1} \quad (A.13b)$$

Here eqns. (A.13a) and (A.13b) hold for series and parallel adaptors respectively. For these cases, λ_n is made equal to unity. Port n thus becomes reflection free.

A.8 REALISABILITY OF LADDER DIGITAL STRUCTURE

For a ladder WDF, each adaptor 'absorbs' one reactive element only (i.e. L or C). Therefore third port resistance can be arbitrary chosen so as to make it if reflection free. Third port of the last adaptor is terminated in a resistance, hence it is an open circuit. This gives the computability of any ladder wave digital filter.

A.9 SCATTERING MATRIX FOR LATTICE WDF

For the symmetric lattice of Fig. (1.4), the voltage and current vectors are related by

$$\begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} Z_a + Z_b & Z_b - Z_a \\ Z_b - Z_a & Z_a + Z_b \end{bmatrix} \begin{bmatrix} I_1 \\ I_2 \end{bmatrix} \qquad (A.14)$$

or,

$$\underline{V} = [Z]\,\underline{I}$$

Using the wave parameters, we write

$$\underline{A} = \underline{V} + R[U]\,\underline{I}$$

$$\underline{B} = \underline{V} - R[U]\,\underline{I}$$

where, $\underline{A} = [A_1 \;\; A_2]^t$, $\underline{B} = [B_1 \;\; B_2]^t$, $R_1 = R_2 = R$

$[U]$ denotes the identity matrix. $\underline{V}$ and $\underline{I}$ are related to $\underline{A}$ and $\underline{B}$ by

$$\underline{V} = 1/2[\underline{A} + \underline{B}], \quad \underline{I} = 1/2R[\underline{A} - \underline{B}] \qquad (A.15)$$

Substituting eqn. (A.15) in (A.14) and rearranging the terms, we get

$$\underline{B} = -([U] + [Z]/R)^{-1}\,([U] - [Z]/R)\,\underline{A}$$

$$= [S]\,\underline{A}$$

Expanding $[Z]$ and simplifying, we find that

$$s_{21} = (Z_b - Z_a)\,R/(Z_a + R)\,(Z_b + R) \qquad (A.16)$$

If S_a and S_b are defined as

$$S_a = (Z_a - R)/(Z_a + R), \quad S_b = (Z_b - R)/(Z_b + R)$$

Then it is easily shown that

$$s_{21} = 1/2(s_b - s_a)$$

A.10 GRAY AND MARKEL'S ALGORITHM

For H(z) as in eqn. (4.1), the algorithm is as follows.

1.1 Let $m := M$

1.2 $ZB_m(z) := A_m(1/z)z^{-m}$; $A_m(z)$ written with reversed order of coefficients

1.3 $k_{m-1} := a_{m,m}$; Last coefficient of $A_m(z)$

1.4 $V_m := p_{m,m}$; Last coefficient of $P_m(z)$

1.5 $A_{m-1}(z) := [A_m(z) - k_{m-1}\, ZB_m(z)]/(1-k^2_{m-1})$

1.6 $P_{m-1}(z) := P_m(z) - ZB_m(z)\, V_m$

1.7 $m := m - 1$

1.8 Go to Step 1.1 if $m >= 0$

$V_o = 1$ and $P_o = 1$ are defined. Because of Step 1.5 $A_{m,o}$ is unity for all m. Using recursion on Step 1.6 and adding all terms we get eqn. (4.2).

A.11 THE LATTICE AND LADDER STRUCTURES

From step 1.2 of (A.10), we get

$$A_m(z) = [A_{m+1}(z) - k_m ZB_{m+1}(z)]/(1-k_m^2) \qquad (A.17)$$

$$A_m(1/z) = [A_{m+1}(1/z) - k_m z^{-1} B_{m+1}(1/z)]/(1-k_m^2)$$

Step 1.6 of (A.10) then gives

$$ZB_{m+1}(z) = B_m(z)\ (1-k_m^2) + k_m\ A_{m+1}(z) \qquad (A.18)$$

Substituting eqn. (A.18) in (A.17) we obtain A_{m+1} as

$$A_{m+1}(z) = A_m(z) + k_m\ B_m(z) \qquad (A.19)$$

Once again using (A.18) and (A.19) we get

$$B_{m+1}(z) = z^{-1}\ k_m\ A_m(z) + z^{-1}\ B_m(z) \qquad (A.20)$$

Eqns. (A.19) and (A.20) give the lattice structure relations. A rearrangement of terms in eqn. (A.19) gives

$$A_m(z) = A_{m+1}(z) - k_m\ B_m(z) \qquad (A.21)$$

Eqns. (A.18) and (A.21) are used to realise the ladder structure.

A.12 THE THROUGHPUT RATE

If T is the total time for one clock cycle then r = 1/T is the throughout rate of the structure. In other words, it is defined as the maximum number of bit parallel input sequences that can be processed in one clock cycle by the processing elements.

A.13 ORTHOGONAL FILTER ALGORITHM

The algorithm for pipelined orthogonal filter [7] is as follows. If T(z) is denoted by as in eqn. (5.20) and we have

$$\phi_i = \begin{bmatrix} D_{oi} & D_{1i} & \cdots & D_{ni} \\ \underline{N}^E_{oi} & \underline{N}^E_{1i} & \cdots & \underline{N}^E_{ni} \end{bmatrix}$$

where $$Q_j(z) = \sum_{i=o}^{n} D_{ji} z^{-i}$$

$$\begin{bmatrix} P_j(z) \\ R_j(z) \end{bmatrix} = \sum_{i=o}^{n} \underline{N}^E_{ji} z^{-i}$$

then we proceed as follows

1.1 Let $i = o$

1.2 Find $\underline{k}_i = [k_{i1}\ k_{i2}]^t$ such that $\underline{N}^E_{oi} = D_{oi}\,\underline{k}_i$

1.3 $$\theta_i = \begin{bmatrix} (1 - \underline{k}_i^{t*}\underline{k}_i)^{-1/2} & -(1 - \underline{k}_i^t\,\underline{k}_i)^{-1/2}\,\underline{k}_i^t \\ -(1 - \underline{k}_i\,\underline{k}_i^{t*})^{-1/2}\,\underline{k}_i & (1 - \underline{k}_i\,\underline{k}_i^{t*})^{-1/2} \end{bmatrix}$$

1.4 Obtain $\theta_i\,\phi_i$ to get

$$\theta_i\,\phi_i = \begin{bmatrix} D_{o,i+1} & D_{1,i+1} & \cdots & D_{n,i+1} \\ 0 & \underline{N}^E_{1,i+1} & \cdots & N^E_{n,i+1} \end{bmatrix}$$

1.5 Shift top row to right and delete left column to get ϕ_{i+1}.

1.6 $i = i + 1$. <u>Go back to Step 1.2</u> unless $i = n - 1$.

APPENDIX B

SIMULATION RESULTS

This appendix shows frequency responses for various low sensitivity implementations of a given transfer function. As an illustration, a third order Butterworth normalised lowpass filter is considered. Its reference domain transfer function is given as

$$H(\psi) = \frac{0.5}{(\psi^3 + 2\psi^2 + 2\psi + 1)}$$

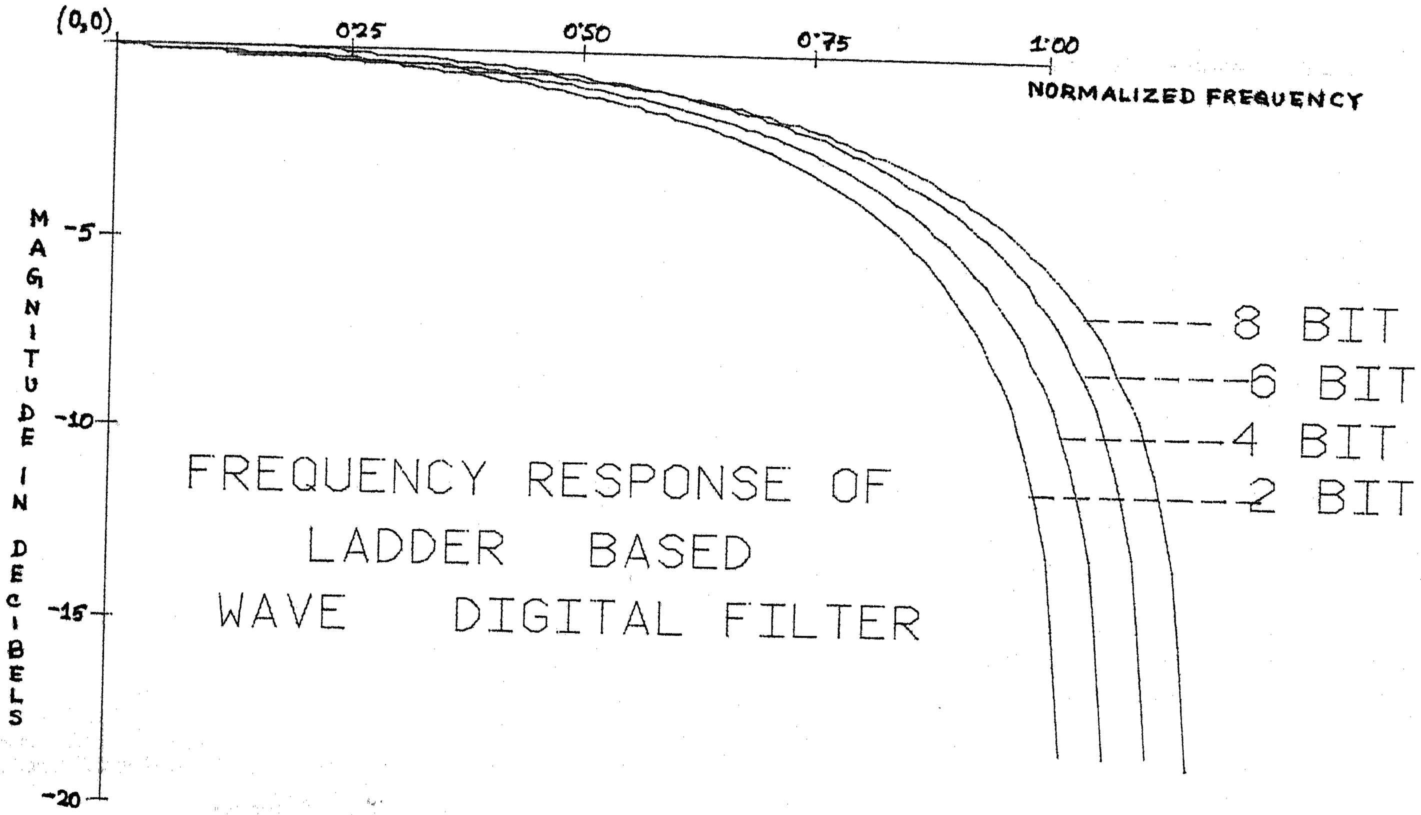
For the direct realisation methods discussed in section 4, the z-domain transfer function using eqn. (3.1a) is obtained as

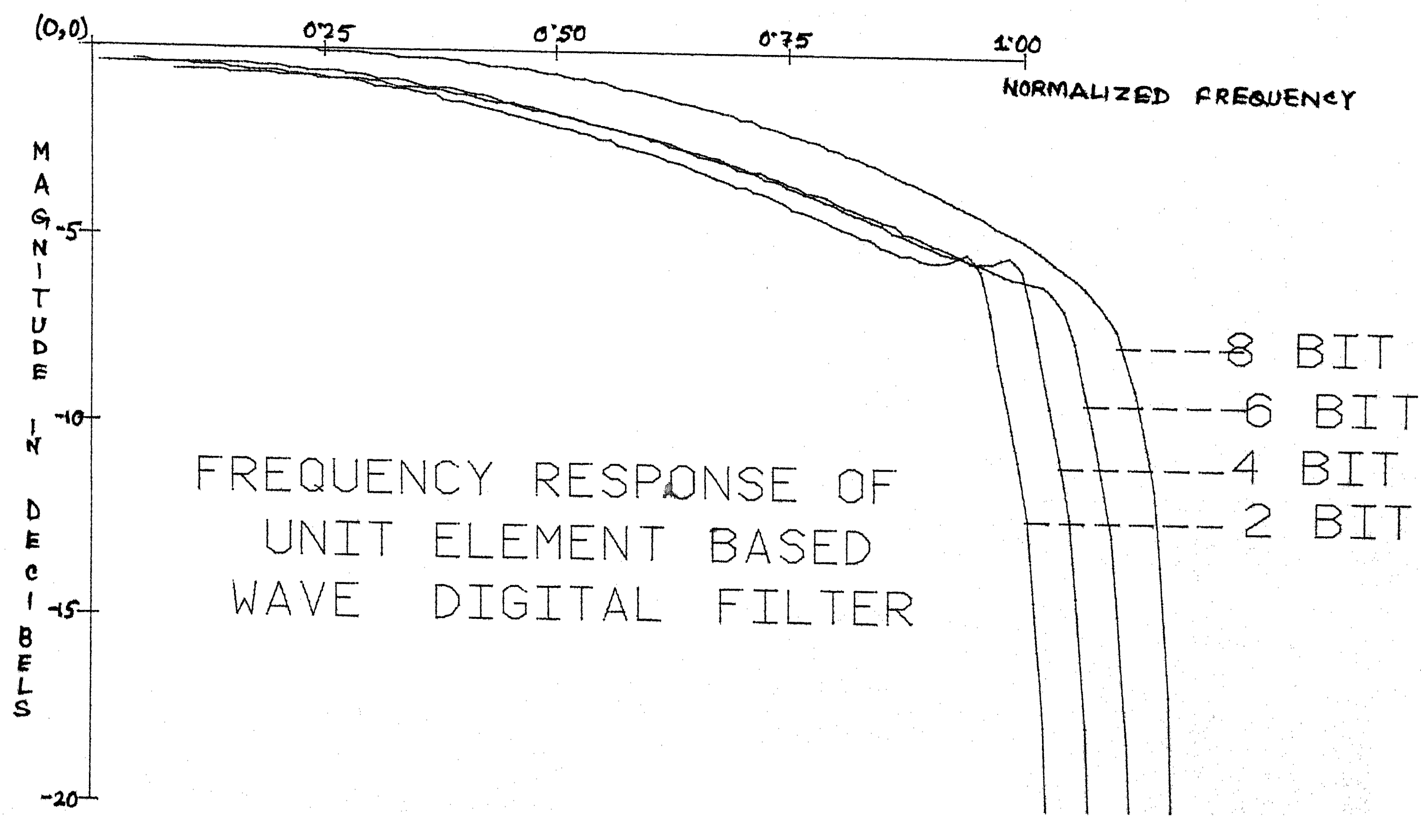
$$H(z) = \frac{1 + 3z^{-1} + 3z^{-2} + z^{-3}}{4(3 + z^{-2})}$$

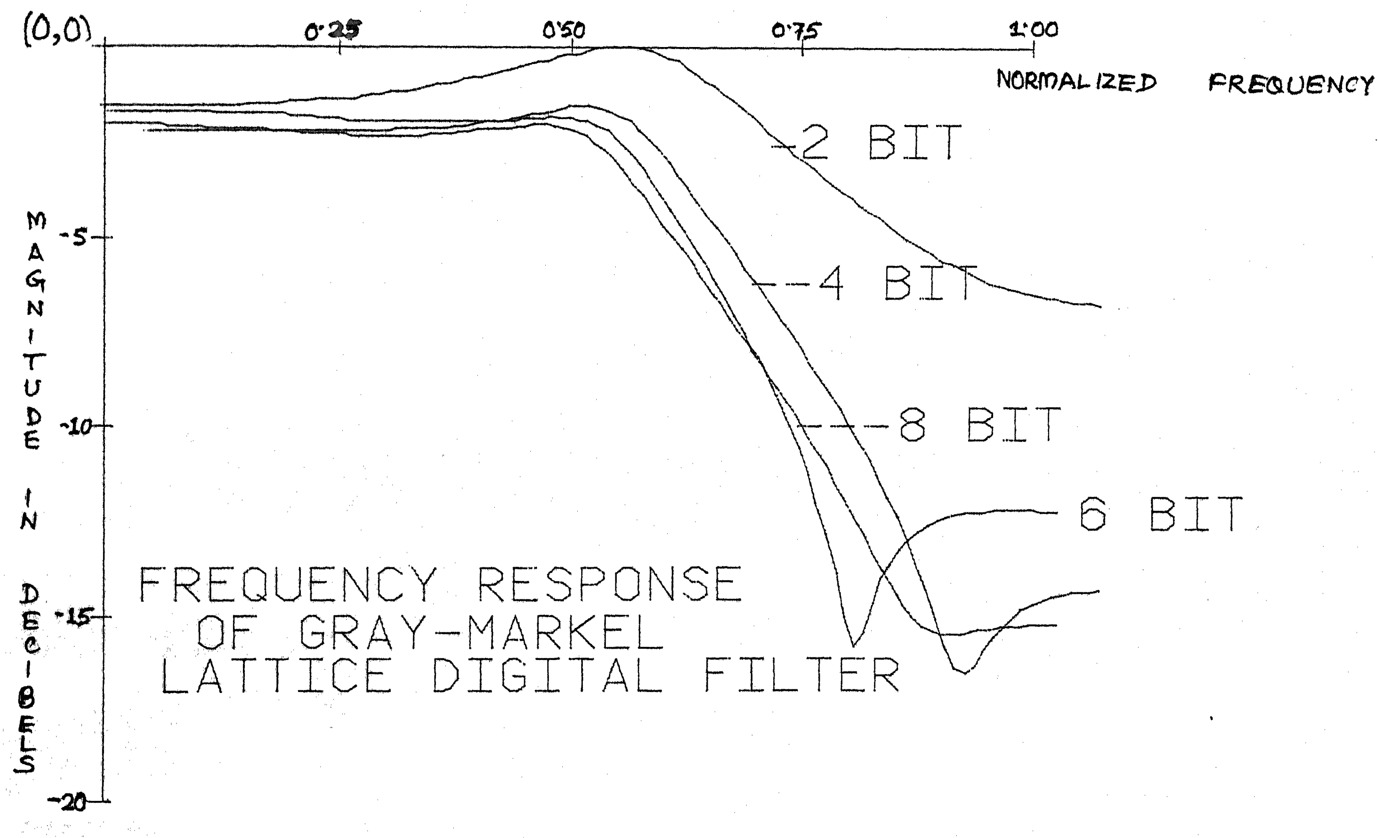
For quantization of the design parameters, number of bits used are 8, 6, 4 and 2 respectively. For comparison, a conventional direct form implementation, using coefficients as multipliers, is also included. All computer programs are listed in Appendix C.

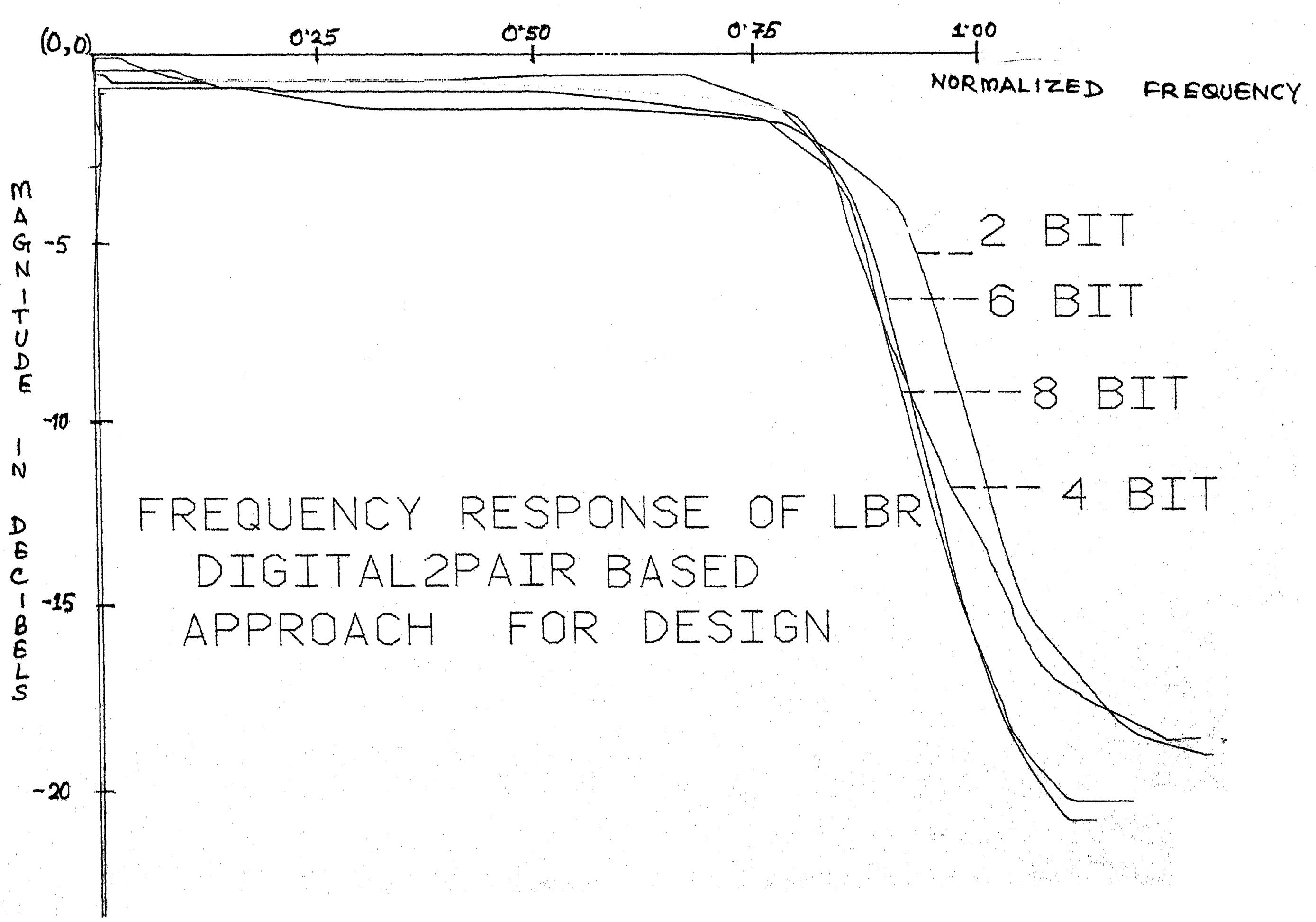
As a variation, for Gray-Markel lattice filter, a third order Chebyshev normalised lowpass filter is considered having

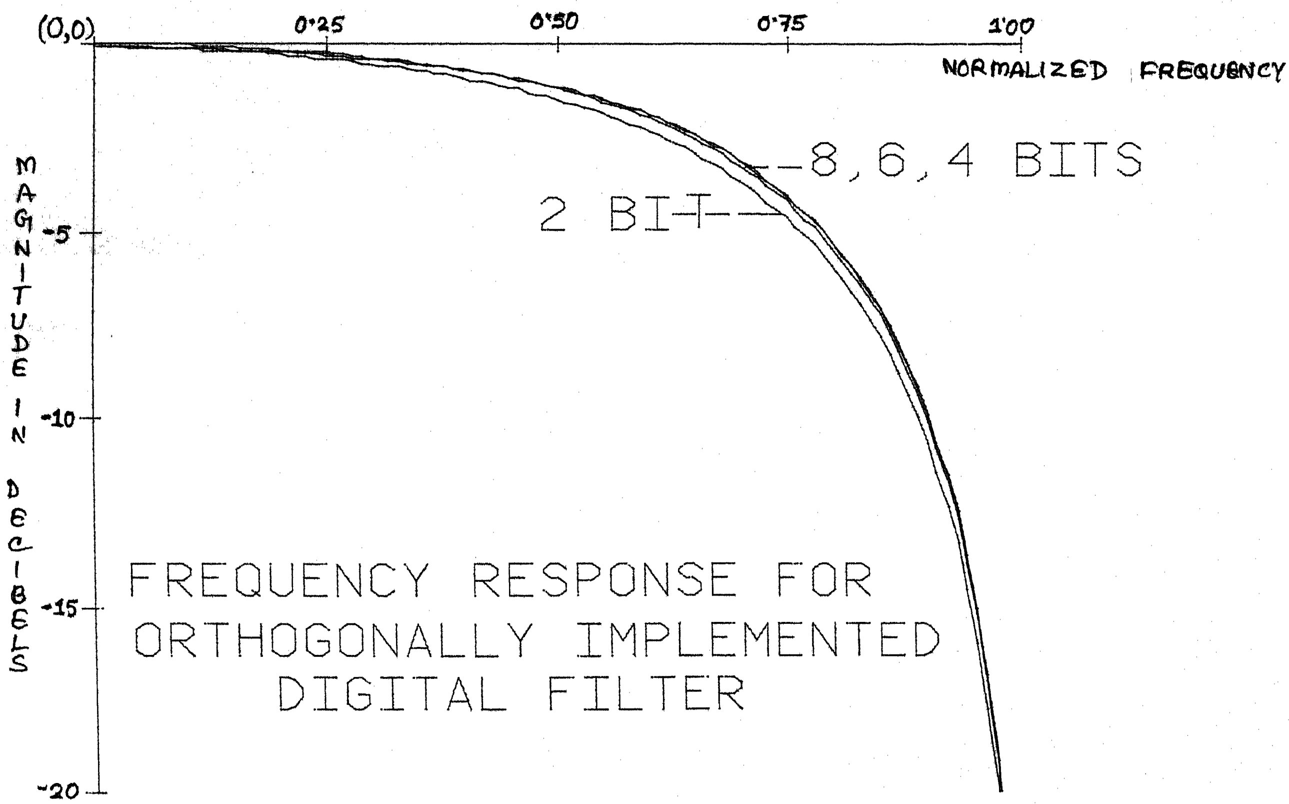
$$H(z) = \frac{1 + 3z^{-1} + 3z^{-2} + z^{-3}}{(7.5 - 2.5z^{-1} + 4.5z^{-2} - 1.5z^{-3})}$$


(0,0)
0·25
0·50
0·75
1·00
NORMALIZED FREQUENCY
MAGNITUDE IN DECIBELS
-5
-10
-15
-20
8 BIT
6 BIT
4 BIT
2 BIT
FREQUENCY RESPONSE OF
LADDER BASED
WAVE DIGITAL FILTER


(0,0)
0·25
0·50
0·75
1·00
NORMALIZED FREQUENCY
MAGNITUDE IN DECIBELS
-5
-10
-15
-20
8 BIT
6 BIT
4 BIT
2 BIT
FREQUENCY RESPONSE OF
UNIT ELEMENT BASED
WAVE DIGITAL FILTER


(0,0)
0'25
0'50
0'75
1'00
NORMALIZED FREQUENCY
MAGNITUDE IN DECIBELS
-5
-10
-15
-20
2 BIT
4 BIT
8 BIT
6 BIT
FREQUENCY RESPONSE
OF GRAY-MARKEL
LATTICE DIGITAL FILTER


(0,0)
0'25
0'50
0'75
1'00
NORMALIZED FREQUENCY
MAGNITUDE IN DECIBELS
-5
-10
-15
-20
2 BIT
6 BIT
8 BIT
4 BIT
FREQUENCY RESPONSE OF LBR
DIGITAL2PAIR BASED
APPROACH FOR DESIGN



(0,0)
0·25
0·50
0·75
1·00
NORMALIZED FREQUENCY
8, 6, 4 BITS
2 BIT
MAGNITUDE IN DECIBELS
-5
-10
-15
-20
FREQUENCY RESPONSE FOR ORTHOGONALLY IMPLEMENTED DIGITAL FILTER

```
C                        FREQUENCY RESPONSE OF
C              CONVENTIONAL DIRECTLY IMPLEMENTED DIGITAL FILTER

        dimension A(10),B(10),H(8,100),FR(100)
        complex ZINV,NUM,DEN
C       Input Format:
C       NP=No. of points,NDEG= degree of trans.function.
C       A(z)= num.of trans. function; B(z)=denominator
        PI=4.*atan(1.)
        read (20,*) NP,NDEG
        do 1 NBT=8,2,-2
        DW=PI/(1.*NP);  T=1./(1.*2**(NBT+1))
        read (20,*),(A(I),I=1,NDEG+1);  read (20,*), (B(I),I=1,NDEG+1)
        TMP=B(1)
        do 10 I=1,NDEG+1
                A(I)=(ifix((A(I)/TMP+T)*2**NBT)*1.)/(1.*2**NBT)
                B(I)=(ifix((B(I)/TMP+T)*2**NBT)*1.)/(1.*2**NBT)
10      continue
        do 20 J=1,NP
                W=(J-1)*DW ;    ZINV=cos(W)*(1.0,0.)-sin(W)*(0.,1.)
                NUM=cmplx(A(1),0.);DEN=cmplx(B(1),0.)
                do 30 I=2,NDEG+1
                NUM=NUM+A(I)*ZINV**(I-1);DEN=DEN+B(I)*ZINV**(I-1)
30              continue
                H(NBT,J)=abs(NUM/DEN)   ;       FR(J)=W/PI
20      continue
1       continue
        do 12 I=1,NP
                write (21,*),(FR(I),H(8,I),H(6,I),H(4,I),H(2,I))
12      continue
        stop
        end
C       ---------------------------------------------------------------
C               LADDER / UNIT ELEMENT BASED WAVE DIGITAL FILTER

        complex ZE,ONE,FA,FB,FC,FD,SA,DB,SC,SD
        dimension NTYPE(10),LC(10),G1(10),G2(10),G3(10),NAME(10)
        dimension H(10,100),FR(100)
C       input format :
C       NP= no. of points;NBT=no. of bits used;NE=no. of 'blocks'
C       NAME  =  'ad' for adapter ; 'ue' for unit element
C       NTYPE = +1 for series adapter ; -1 for parallel adapter
C       LC    = +1 for capacitor ; -1 for inductor (3rd adapter port)
C       T1,T2,T3 :Res./admittances for series/parallel adaptors.
        PI=4.0*atan(1.0);ONE=(1.0,0.0)
        read (20,*) NP,NE
        do 11 NBT=8,2,-2
        TMP=1.0/2**(NBT+1)
        do 10 I=1,NE
                read (20,100) NAME(I)
100             format(a2)
                if (NAME(I).eq.'AD') go to 20
                read (20,*) R
                go to 10
20              read (20,*),NTYPE(I),LC(I),T1,T2,T3
                Y1=2.0*T1/(T1+T2+T3);Y2=2.0*T2/(T1+T2+T3)
                G1(I)=(ifix((Y1+TMP)*2**NBT)*1.0)/2**NBT
                G2(I)=(ifix((Y2+TMP)*2**NBT)*1.0)/2**NBT
                G3(I)=2.0-G1(I)-G2(I)
10      continue
```

```
      DW=PI/NP
21    do 25 J=1,NP
            W=(J-1)*DW ; R1=cos(W); R2=sin(W);ZE=cmplx(R1,R2)
            FA=ONE; FB=(0.0,0.0);FD=ONE;FC=FB
            do 40 I=NE,1,-1
                  if (NAME(I).eq.'AD') go to 30
                  call UNITEL(SA,SB,SC,SD,ZE)
                  go to 35
30                call ADAPTR(NTYPE(I),LC(I),G1(I),G2(I),G
     13(I),SA,SB,SC,SD,ZE)
35                FA=SA*FA+SB*FC;FB=SA*FB+SB*FD
                  FC=SC*FA+SD*FC;FD=SC*FA+SD*FC
40          continue
            H(NBT,J)=ABS(ONE/FA);FR(J)=W/PI
25    continue
11    continue
      do 12 I=1,NP
      WRITE (21,*),(FR(I),H(8,I),H(6,I),H(4,I),H(2,I))
12    continue
      stop
      end
C     -----------------------------------------------------------------
C          FREQUENCY RESPONSE OF LATTICE WAVE DIGITAL FILTER

      dimension NTYPE(10),LC(10),G1(10),G2(10),G3(10),NAME(10),FR(100)
      dimension HH(8,100)
      complex   ZE,ONE,FA,FB,FC,FD,SA,SB,SC,SD,H(2,100)
      PI=4.0*atan(1.0); ONE=(1.0,0.0)
      read (20,*) NP
      DW=PI/(NP*1.0)
      do 1 NBT=8,2,-2
      TMP=1.0/2**(NBT+1)
      do 50 LOOP=1,2
            read (20,*) NE,NT,NM
            do 10 I=1,NE
                  read (20,100) NAME(I)
100               format(a2)
                  if (NAME(I).eq.'AD') go to 20
                  read (20,*) R
                  go to 10
20                read (20,*),NTYPE(I),LC(I),T1,T2,T3
                  Y1=2.0*T1/(T1+T2+T3) ; Y2=2.0*T2/(T1+T2+T3)
                  G1(I)=(ifix((Y1+TMP)*2**NBT)*1.0)/2**NBT
                  G2(I)=(ifix((Y2+TMP)*2**NBT)*1.0)/2**NBT
                  G3(I)=2.0-G1(I)-G2(I)
10          continue
            do 25 J=1,NP
                  W=1.0*(J)*DW
                  R1=cos(W); R2=sin(W)    ; ZE=cmplx(R1,R2)
                  FA=ONE; FB=(0.0,0.0) ;          FD=ONE; FC=FB
                  do 40 I=NE,1,-1
                        if (NAME(I).eq.'AD') go to 30
                        call UNITE(SA,SB,SC,SD,ZE)
                        go to 35
30                      call ADAPTR(NTYPE(I),LC(I),G1(I)
     1                  ,G2(I),G3(I),SA,SB,SC,SD,ZE)
35                      FA=SA*FA+SB*FC;FB=SA*FB+SB*FD
                        FC=SC*FA+SD*FC;FD=SC*FB+SD*FD
40                continue
                  if (NT.eq.1) go to 22
```

```
                        H(LOOP,J)=(FC+FD*NM)/(FA+FB*NM)
                        go to 25
22                      H(LOOP,J)=(FC+FD*NM/ZE)/(FA+FB*NM/ZE)
25             continue
50    continue
      do 27 J=1,NP
             FR(J)=J*1.0/NP ;       HH(NBT,J)=abs(H(2,J)-H(1,J))
27    continue
1     continue
      do 12 I=1,NP
      write (21,*),(FR(I),(HH(NBT,I),NBT=8,2,-2))
12    continue
      stop
      end
C     ---------------------------------------------------------------
      SUBROUTINE ADAPTR(NT,K,X1,X2,X3,A,B,C,D,Z)
      complex DN,C1,C2,C3,C4,A,B,C,D,Z
      DN=1.0-NT*K*(1.0-X3)/Z
      C1=NT*(1.0-X1)+K*(1.0-X3)/Z  ;   C2=-X1*(NT*1.0-K/Z)
      C3=-X2*(NT*1.0-K/Z) ; C4=NT*(1.0-X2)+K*(1.0-X3)/Z
      A=DN/C3;B=-C4/C3;C=C1/C3;D=(C2*C3-C1*C4)/(DN*C3)
      return
      end

      SUBROUTINE UNITEL (A,B,C,D,Z)
      complex A,B,C,D,Z
      A=sqrt(Z);D=ONE/A  ; B=(0.0,0.0);C=B
      return
      end
      ---------------------------------------------------------------
        {     GRAY AND MARKEL ALGORITHM FOR LATTICE DIGITAL FILTER     }

PROGRAM GRAYMARKEL(INPUT,OUTPUT);
  var
      M,N,I,J:integer;
      TMP:real;
      K,V:array[0..10] of real;
      A,P,ZB:array[0..10,0..10] of real;
  begin
      readLN M;
      for N:=0 to M do
          read P[M,N];
      readLN;
      for N:=0 to M do
          read A[M,N];
      readLN;
      TMP:=A[M,0];
      for N:=0 to M do
      begin
          A[M,N]:=A[M,N]/TMP;     P[M,N]:=P[M,N]/TMP
      end;
      for I:=M downto 1 do
      begin
          for J:=0 to I do            ZB[I,J]:=A[I,I-J];
          K[I-1]:=A[I,I];
          for J:=0 to I-1 do
               A[I-1,J]:=(A[I,J]-K[I-1]*ZB[I,J])/(1.0-K[I-1]*K[I-1]);
          V[I]:=P[I,I];
          for J:=0 to I-1 do          P[I-1,J]:=P[I,J]-ZB[I,J]*V[I]
      end;
```

```
      V[0]:=P[0,0];
      WRITELN ( 'K-parameters are');
      for N:=0 to M-1 do          WRITE ( ' k( ',N:1, ')= ',K[N]:10:6);
      WRITELN;
      WRITELN( 'Tap parameters are');
      for N:=0 to M do
          WRITE( ' v( ',N:1, ')= ',V[N]:10:6)
  end.
C     -------------------------------------------------------------
C       FREQUENCY RESPONSE OF GRAY AND MARKEL LATTICE DIGITAL  FILTER

      dimension A(10),ZBM(10,10),H(8,100),FR(100),RM(10),V(10)
      complex ZINV,NUM,DEN,RM
      PI=4.*atan(1.) ; read (20,*) NP,NDEG
      do 1 NBT=8,2,-2
      DW=PI/(1.*NP)
      T=1./(1.*2**(NBT+1))
      read (20,*),(A(I),I=1,NDEG+1) ; read (20,*),(V(J),J=1,NDEG+1)
      read (20,*), ((ZBM(J,I),I=1,NDEG+1),J=1,NDEG+1)
      TMP=A(1)
      do 10 I=1,NDEG+1
      do 15 J=1,NDEG+1
              ZBM(J,I)=(ifix((ZBM(J,I)/TMP+T)*2**NBT)*1.)/(1.*2**NBT)
15    continue
              A(I)=(ifix((A(I)/TMP+T)*2**NBT)*1.)/(1.*2**NBT)
10    continue
      do 20 JJ=1,NP
              W=(JJ-1)*DW ; ZINV=cos(W)*(1.0,0.)-sin(W)*(0.,1.)
              DEN=cmplx(A(1),0.)
      do 16 J=1,NDEG+1
      RM(J)=CMPLX(ZBM(J,1),0.0)
16    continue
              do 30 I=2,NDEG+1
      do 17 J=1,NDEG+1
                      RM(J)=RM(J)+ZBM(J,I)*ZINV**(I-1)
17    continue
                      DEN=DEN+A(I)*ZINV**(I-1)
30            continue
      NUM=(0.0,0.0)
      do 18 J=1,NDEG+1
      NUM=NUM+RM(J)*V(J)
18    continue
              H(NBT,JJ)=abs(NUM/DEN) ;        FR(JJ)=W/PI
20    continue
1     continue
      do 12 I=1,NP
              write (21,*),(FR(I),H(8,I),H(6,I),H(4,I),H(2,I))
12    continue
      stop
      end
C     -------------------------------------------------------------
C         FREQUENCY RESPONSE OF LBR 2-PAIR BASED DIGITAL FILTER

      complex ZE,ONE,FA,FB,FC,FD,SA,DB,SC,SD
      dimension NAME(10),SIG(10),ALP(10),K(10),H(10,100),FR(100)
      PI=4.0*atan(1.0) ; ONE=(1.0,0.0)
      read (20,*) NP,NE,R
      do 11 NBT=8,2,-2
      TMP=1.0/2**(NBT+1);TR=(IFIX((R+TMP)*2**NBT)*1.0)/(2**NBT)
      do 10 I=1,NE
```

```
                read (20,100) NAME(I)
100             format(a3)
                if (NAME(I).eq.'TWO') go to 20
                read (20,*) K(I),SIG(I)
                go to 5
20              read (20,*),K(I),SIG(I),ALP(I)
                ALP(I)=(ifix((ALP(I)+TMP)*2**NBT)*1.0)/2**NBT
5               SIG(I)=(ifix((SIG(I)+TMP)*2**NBT)*1.0)/2**NBT
10        continue
          DW=PI/NP
          do 25 J=1,NP
                W=(J-1)*DW
                R1=cos(W); R2=sin(W)
                ZE=cmplx(R1,R2)
                FA=ONE; FB=(0.0,0.0)
                FD=ONE; FC=(0.0,0.0)
                do 40 I=NE,1,-1
                      if (NAME(I).eq.'TWO') go to 30
                      call LBRONE(K(I),SIG(I),SA,SB,SC,SD,ZE)
                      go to 35
30                    call LBRTWO(K(I),SIG(I),ALP(I),SA,SB,SC,SD,ZE)
35                    FA=SA*FA+SB*FC; FB=SA*FB+SB*FD; FC=SC*FA+SD*FC
                      FD=SC*FB+SD*FD
40              continue
                H(NBT,J)=ABS((FC+FD*TR)/(FA+FB*TR)); FR(J)=W/PI
25        continue
11        continue
          do 12 I=1,NP
          WRITE (21,*),(FR(I),H(8,I),H(6,I),H(4,I),H(2,I))
12        continue
          stop
          end

          SUBROUTINE LBRONE(L,SG,A,B,C,D,Z)
          complex A,B,C,D,Z
          A=(1.0+SG/Z)/1.0
          B=L*(1.0-SG)/(Z*1.0)
          C=L*(1.0-SG)/1.0
          D=(SG+1.0/Z)/1.0
          return
          end

          SUBROUTINE LBRTWO(L,SG,BET,A,B,C,D,Z)
          complex A,B,C,D,Z
          A=(1.0+BET*(1.0+SG)/Z+SG/(Z*Z))/1.0
          B=L*(1.0-SG)*(BET*1.0+1.0/Z)/(Z*1.0)
          C=L*(1.0-SG)*(1.0+BET/Z)/1.0
          D=(SG*1.0+BET*(1.0+SG)/Z+1.0/(Z*Z))/1.0
          return
          end
C         -------------------------------------------------------------
            {             DESIGN OF ORTHOGONAL DIGITAL FILTER             }

PROGRAM VLSI(INPUT,OUTPUT);
  const
      U=2;
  type
      Y=array[0..2,0..5] of real;
  var
      I,J,M,N,NDEG,P:integer;
```

```
     K1,K2,TMP1,TMP2:real;
     T,LAMBDA,TL,TTL:Y;
 procedure MATPRO(M,P,N:integer;var A,B,C:Y);
     var
         I,J,K:integer;
     begin
         for I:=0 to M do
               for J:=0 to N do
                 begin
                     C[I,J]:=0.0;
                     for K:=0 to P do
                           C[I,J]:=C[I,J]+A[I,K]*B[K,J]
                 end
     end;
 begin
     readLN NDEG;
     for I:=0 to 2 do
     begin
         for J:= 0 to NDEG do
               read LAMBDA[I,J];
         readLN
     end;
     WRITELN;
     WRITELN;
     WRITELN;

     WRITE ( 'Coefficient values for Pipelined Orthogonal  ');
     WRITELN ( 'Digital Filter : Degree = ',NDEG:1);
     for I:=0 to NDEG do
     begin
         K1:=LAMBDA[1,0]/LAMBDA[0,0];
         if K1>1.0 then K1:=1.0;
         if K1<-1.0 then K1:=-1.0;
         TMP1:=ABS(SQRT(1.0-K1*K1));
         T[0,0]:=1.0/TMP1;T[0,1]:=-K1/TMP1;T[0,2]:=0.0;
         T[1,0]:=-K1/TMP1;T[1,1]:=1.0/TMP1;T[1,2]:=0.0;
         T[2,0]:=0.0;T[2,1]:=0.0;T[2,2]:=1.0;
         MATPRO(U,U,NDEG,T,LAMBDA,TL);
         K2:=TL[2,0]/TL[0,0];
         if K2>1.0 then K2:=1.0;
         if K2<-1.0 then K2:=-1.0;
         TMP2:=ABS(SQRT(1.0-K2*K2));
         T[0,0]:=1.0/TMP2;T[0,1]:=0.0;T[0,2]:=-K2/TMP2;
         T[1,0]:=0.0;T[1,1]:=1.0;T[1,2]:=0.0;
         T[2,0]:=-K2/TMP2;T[2,1]:=0.0;T[2,2]:=1.0/TMP2;
         MATPRO(U,U,NDEG,T,TL,TTL);
         WRITELN;
         WRITELN ( 'K(  ',I:1,',1  ) = ',K1:7:4,'  K(  ',I:1,',2  ) =',K2:7:4);
         for J:=0 to NDEG do
               LAMBDA[0,J]:=TTL[0,J];
         for J:=0 to NDEG-1 do
           begin
               LAMBDA[1,J]:=TTL[1,J+1];
               LAMBDA[2,J]:=TTL[2,J+1]
           end;
         LAMBDA[1,NDEG]:=0.0;
         LAMBDA[2,NDEG]:=0.0
     end
 end.
```

REFERENCES

[1] A. Fettweis, ''Digital filter structures related to classical filter networks,'' Digital Signal Processing II, edited by DSP Committee, IEEE ASSP Soc., pp. 475-485, IEEE Press, 1975.

[2] ____________, ''Pseudopassivity, sensitivity and stability of wave digital filters,'' IEEE Trans. Circ. Th., Vol. CT-19, pp. 668-673, Nov. 1972.

[3] A.H. Gray and J.D. Markel,''Digital lattice and ladder filter synthesis,'' IEEE Trans. Audio Electroaccoust., Vol. AU-21, pp. 491-500, Dec. 1973.

[4] ____________, ''A normalised digital filter structure,'' IEEE Trans. Acoust., Speech and Signal Proc., Vol. ASSP-23, pp. 268-277, June 1975.

[5] P. Dewilde and E. Deprettere, ''Orthogonal cascade realisation of real multiport digital filters,'' Int. J. Circ. Th. and Appln., Vol. 8, pp. 245-277, 1980.

[6] P.P. Vaidyanathan and S.K. Mitra, ''Low passband sensitivity digital filters : A generalised viewpoint and synthesis procedures,'' Proc. IEEE, pp. 404-423, April 1984.

[7] S.K. Rao and T. Kailath, ''Orthogonal digital filters for VLSI implementation,'' IEEE Trans. Circ. and Sys., Vol. CAS-31, pp. 933-945, Nov. 1984.

[8] P.P. Vaidyanathan, ''Low sensitivity filters, wave digital filters and orthogonal digital filters : a unified framework, based on LBR approach,'' IEEE Int. Conf. on ASSP, pp. 537-541, Bangalore 1984.

[9] D.S. Humpherys, ''The analysis, design and synthesis of Electrical filters,'' Prentice Hall, 1970.

[10] H.J. Orchard, ''Inductorless filters,'' Electron. Lett., Vol. 2, No. 6, pp. 224-225, June 1966.

[11] D.E. Johnson, ''Introduction to filter theory,'' Prentice Hall 1976.

[12] G.C. Temes and S.K. Mitra (ed.), ''Modern filter and design,'' pp. 276-289, John Wiley, 1973.

[13] K. Renner and S.C. Gupta, ''On the design of wave digital filters with low sensitivity property,'' IEEE Trans. Circ. Th., Vol. CT-20, pp. 555-566, Sept. 1973.

[14] A. Antoniou, ''Digital Filters : Analysis and design,'' pp. 89, McGraw Hill, 1979.

[15] G.C. Srivastava, ''Design of Fluidic Filters,'' M.Tech. Thesis, I.I.T. Kanpur, 1986.

[16] B.V. Limaye, ''Functional Analysis,'' Wiley-Eastern, 1981.

[17] A. Antoniou, ''Digital Filters : Analysis and design,'' McGraw Hill, 1979.

[18] A. Fettweis, ''Wave digital filters : theory and practice,'' Proc. IEEE, Vol. 74, pp. 270-327, Feb. 1986.

[19] A. Fettweis and K. Meerkotter, ''On adaptors for wave digital filters,'' IEEE Trans. Acoust., Speed and Signal Proc., Vol. ASSP-23, pp. 516-525, Dec. 1975.

[20] A. Fettweis, ''Canonic realisation of ladder wave digital filters,'' Int. J. Circ. Th. and Appln., Vol. 3, pp. 321-332, Dec. 1975.

[21] K. Renner and S.C. Gupta, ''On the design of wave digital filters with a minimal number of multipliers,'' IEEE Trans. Circ. and Sys., Vol. CAS-21, pp. 137-145, Jan. 1974.

[22] A. Fettweis, H. Levin and A. Sedlemeyer, ''Wave digital lattice filter,'' Int. J. Circ. Th. and Appln., Vol. 2, pp. 203-211, June 1974.

[23] A. Fettweis, ''Digital circuits and systems,'' IEEE Trans. Circ. and Sys., Vol. CAS-31, pp. 31-48, Jan. 1984.

[24] A.H. Gray, Jr., ''Passive cascaded lattice digital filters,'' IEEE Trans. Circ. and Sys., Vol. CAS-27, pp. 337-344, May 1980.

[25] R.W. Newcomb, ''Linear multiport synthesis,'' pp. 305-309, McGraw Hill, 1966.

[26] E. Deprettere, P. Dewilde and Prabhakar Rao, ''Orthogonal filter design and VLSI implementation,'' IEEE Int. Conf. on ASSP, pp. 779-780, Bangalore, 1984.

[27] J.E. Volder, ''The CORDIC trigonometric computing technique,' IRE Trans. Electronic Computers, Vol. EC-8, pp. 330-334, 1959.

[28] C.A. Mead and L.A. Conway, ''Introduction to VLSI systems,'' Addison-Wesley, 1980.

[29] S.K. Rao and T. Kailath,''Pipelined Orthogonal Digital lattice filters,'' Int. Conf. on ASSP, pp. 11.10.1-11.10.4, 1984.

A 98320

EE-1987-M-RAS-LOW